NASA Conference Publication 3034
Part 2

Computational Methods for Structural Mechanics and Dynamics

Edited by
W. Jefferson Stroud
Jerrold M. Housner
John A. Tanner
and Robert J. Hayduk
NASA Langley Research Center
Hampton, Virginia

Proceedings of a workshop sponsored by
the National Aeronautics and Space
Administration and held at
NASA Langley Research Center
Hampton, Virginia
June 19–21, 1985

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1989

PREFACE

This document contains the proceedings of the Workshop on Computational Methods for Structural Mechanics and Dynamics held at NASA Langley Research Center, June 19-21, 1985. The workshop was sponsored by NASA Langley Research Center.

The workshop had two objectives. The first objective was to introduce to the structural analysis technical community a new Langley research activity in structural analysis called Computational Structural Mechanics, or CSM. The second objective was to hear experts discuss important structural analysis problems and methods for solving those problems.

The workshop was organized into the following four sessions:

1.	Local/Global Nonlinear Stress Analysis	-	Full day	-	June 19
2.	Tire Modeling	-	Half day	-	June 20
3.	Transient Dynamics	-	Half day	-	June 20
4.	Multi-Body Dynamics	-	Full day	-	June 21

Each session closed with a panel discussion.

Papers in these proceedings are grouped by session and identified in the contents. The order of the papers is the order of the presentations at the workshop. The proceedings also include any transcription of questions and answers that followed each paper and panel discussions that followed each session.

The use of trade names or names of manufacturers in this publication does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

W. Jefferson Stroud
Jerrold M. Housner
John A. Tanner
Robert J. Hayduk
Workshop Co-Chairmen

CONTENTS

PART 1*

LOCAL/GLOBAL NONLINEAR STRESS ANALYSIS

*Presented under separate cover.

PRECEDING PAGE BLANK NOT FILMED

TIRE MODELING

PART 2

TRANSIENT DYNAMICS

MULTI-BODY DYNAMICS

PART 2

TRANSIENT DYNAMICS

[redacted]

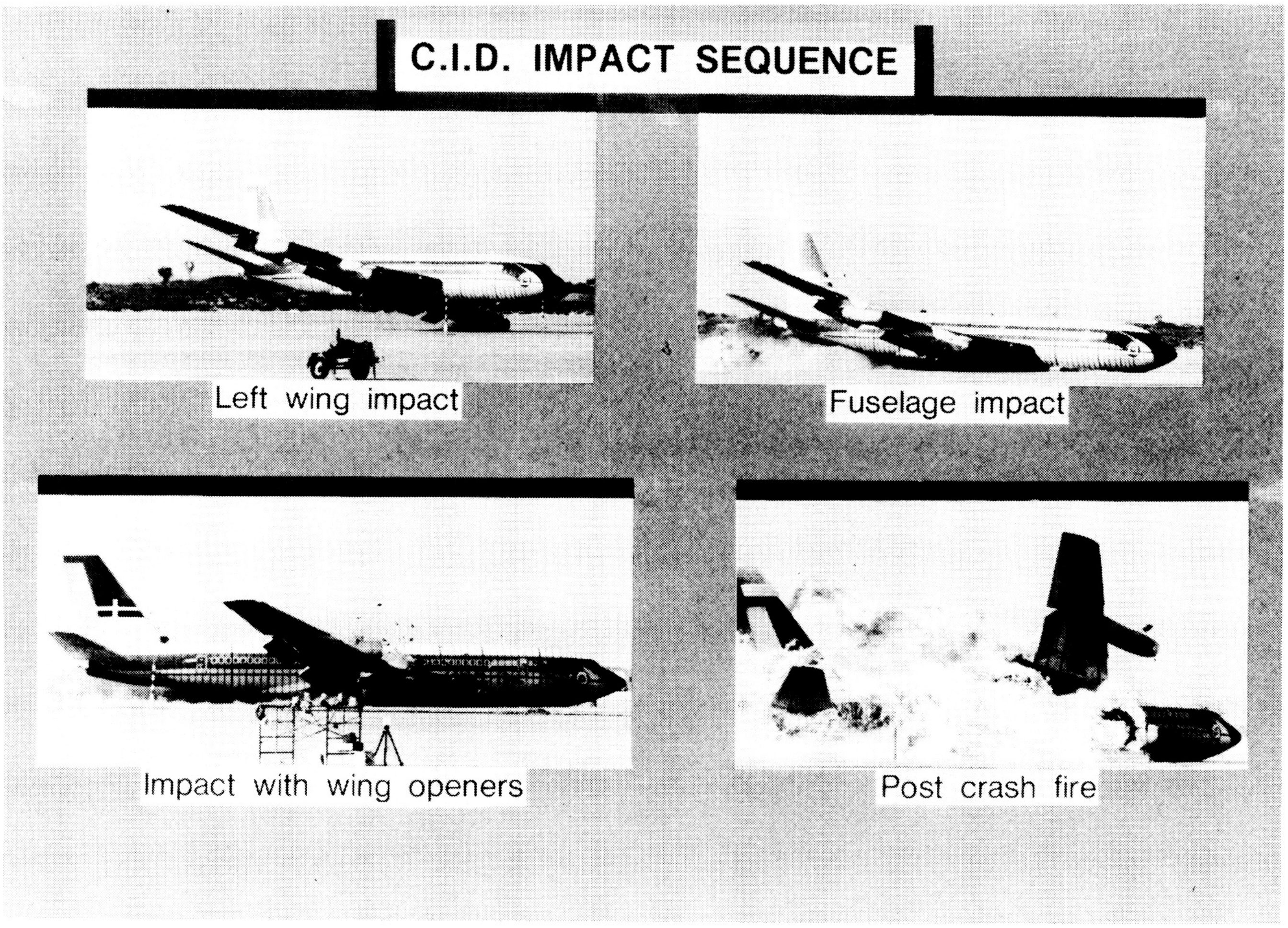
SUMMARY

THURSDAY, JUNE 20 - TRANSIENT DYNAMICS

Robert J. Hayduk, NASA Langley Research Center: We have an interesting session for you this afternoon on transient dynamics. One paper will project opportunities for increasing the efficiency and accuracy of computer codes. Two papers will discuss implementation and experience in time-stepping algorithms, and two papers will give some results on applications of various codes on the scaler and vector machines that are available today.

Everyone has his pet problem in nonlinear structural dynamics. I'm no exception. In order to begin this session, let me show you a problem that is of interest to the CSM group here at Langley and to me. This is a four-sequence photograph of the December 1, 1984 Controlled Impact Demonstration (CID) crash test of a Boeing 720 aircraft. We had planned for the impact to be symmetric, but that isn't what happened. The aircraft touched down with the left wing rolled at approximately 11 degrees with the aircraft nearly level but yawed about 11 to 12 degrees. This caused the aircraft to pitch nose down, and the fuselage to impact on the nose first. We had planned for the aircraft to be slightly pitched up to achieve an initial impact on the aft end of the fuselage.

After about 1.8 seconds, the aircraft slid into some wing cutters, which opened up the number three engine--the inboard engine of the right side of the aircraft. The disintegrating engine caused a huge fire to erupt as you can see from this last photograph. At the time of contact with the wing openers, the aircraft was yawed approximately 38 degrees. Now what we'd like to be able to do--in fact, what we had planned initially to do--was to analyze this impact for the initial portion of the crash scenario--the portion prior to impact with the wing cutter. With the asymetric impact that actually occurred in the test, we have to use a full finite element model that can handle the asymmetric case. Eventually we would like to analyze the longer duration impact with the wing cutters and the slide out beyond, which is about another 2.5 seconds. The initial attempt at analysis with the symmetric half model, which is a very simple beam stringer and membrane model with nonlinear springs of about 220-some elements, and 230 equations, simulating approximately 0.4 second with a full Cyber 175, cost us about 1.4 hours of computer time with the DYCAST computer program. If this problem is scaled up to a machine that would give us perhaps a 10-to-1 increase in computer speed with a full model approximately 1100 masses,



C.I.D. IMPACT SEQUENCE

Left wing impact

Fuselage impact

Impact with wing openers

Post crash fire

4000 elements, and 11,000 equations-to simulate approximately 1 second of real time, using a machine with a very large memory capability, then we're projecting about 18 hours of computer time to solve this problem. And that is without taking advantage of the technology improvements in computational methods. After corrobrating the computer program with the CID experimental data, we plan to do parametric studies of other crash scenarios to eliminate the need for full-scale crash testing of other transport aircraft. These tests are very expensive and actually occur only about once every 20 years. The CID test cost on the order of $10 million to accomplish. We're looking for improvements in analytical capabilities through the CSM activity and your activity to reduce this cost and make the parametric studies feasible.

IMPROVING TRANSIENT ANALYSIS TECHNOLOGY FOR AIRCRAFT STRUCTURES

R. J. Melosh
Duke University, Durham, NC

Mladen Chargin
NASA Ames Research Center, Moffett Field, CA

1. INTRODUCTION

Aircraft dynamic analyses are demanding of computer simulation capabilities. The modeling complexities of semi-monocoque construction, irregular geometry, high-performance materials, and high-accuracy analysis are present. At issue are the safety of passengers and the integrity of the structure for a wide variety of flight-operating and emergency conditions.

Figure 1 is a sketch of a typical structure. It dipicts one of NASA Ames designs of an oblique wing. The wing chord varies from 18.36 inches at the root to 37.8 inches at its 254.4 inch span. The skins are formed of a 0°/±45°/90° 76%/14%/10% graphite/epoxy composite. The skin varies in thickness from .625 inch at the root to .184 inch at the tip. The skins are supported by 5 vertically stiffened spars and 14 stiffened ribs. All the support structure is designed in aluminum. The wing must be proofed against landing, lift and drag, gust, buffet, vibration, and oscillating aerodynamic loading.

The figures and text that follow examine the technology which supports engineering of aircraft structures using computer simulation. They briefly describe available computer support and recommend improving accuracy and efficiency. Improved accuracy of simulation will lead to more economical structure. Improved efficiency will result in lowering development time and expense.

2. SIMULATION SUPPORT

Figure 2 lists the dynamicists' tasks for computer simulation of transient analysis. Dynamicists define the finite-element representation of their structure and its boundary conditions. They select the procedures to use in integrating the equations of motion over time, and define the models and extent of stress evaluation. They interpret analysis results with respect to the real system, drawing upon their knowledge of the models, algorithms, and the computer configuration which implements the simulation.

Figure 3 identifies the computer capabilities which support implementation of the tasks of Figure 2. Existing finite-element models provide for both Rayleigh-Ritz and heuristic models. Three methods of reducing the vector basis, four classes of numerical quadrature, and at least three processes for evaluating stresses are available. Interpretation software facilitates plotting and tabulating data.

3. ACCURACY ASSESSMENT

Figure 4 is typical of the type of data that would be useful to the dynamicist in assessing analysis accuracy. The continuous folded line on this figure plots the actual spatial discretization error for the first two resonant frequencies. The dashed folded line portrays the error predicted using accuracy qualifying logic.

Figure 5 shows similar data qualifying the prediction of transient response with respect to spatial discretization error. The fact that this error can accumulate during the history emphasizes the need for continuous monitoring of this error source.

Figure 6 notes the principal sources of inaccuracy in each of the simulation tasks. The sources include spatial discretization, time discretization, process, round-off, idealization, and human errors. These sources induce accuracy loss in each task which can accumulate from task to task and obliterate accuracy.

Figure 7 is a bar chart of the comprehensiveness of support of each error source in contemporary simulations. No known production computer code is complete with respect to any source. Most codes provide partial protection against process and roundoff error only. Consequently, we cannot regard transient analysis results as reliable. For some of these sources, new technology is needed to determine accuracy; for others, suggested techniques require evaluation; and for the rest, only implementation in production codes is necessary.

4. ANALYSIS EFFICIENCY

Figure 8 cites the sources of inefficiency in simulation tasks with respect to technology and software. These sources involve use of non-optimum models, inappropriate integration algorithms, and unsuitable space and time grids. Lack of efficiency measures in computer codes inhibits experimental improvement of simulation efficiency in practice.

Figure 9 illustrates the inefficiency of available beam models for predicting modal frequencies. This figure shows the logarithmic relation between the number of modal frequencies and the equivalent number of elements and nodal variables. The first is a measure of the computer resources needed for equation coefficient; the second, those needed for equation integration. The data show that the efficiency of the Bernouilli-Euler beam model is less than 50 percent of that of the ideal model.

Figure 10 focuses on the efficiency of nodal siting for the beam. The abscissa of the graph measures the number of calculations. The ordinate indicates the number of accurate modes. These curves illustrate the existence of a distinct optimum grid for each mode. Analysis using the optimum grid requires only one-third the calculations of the average grid.

Figure 11 gives the conventional wisdom for selecting the time integration process of transient analysis. This table pertains to linear dynamic analyses. Considering the number of calculations, the data indicates that a different algorithm is advisable depending upon whether the frequency content of response is high or low and whether the integration time is brief or extended compared with the period of the fundamental mode. Comparing the best to the worst choice of algorithm we find an advantage of a factor which is a function of the order and band of the integration operator matrix.

Figure 12 provides data for comparing the efficiency of integration algorithms for a highly nonlinear transient analysis of a cylinder. These data indicate that explicit (central differences) and explicit (Newmark Beta) are competitive but modal synthesis is not. Choosing the better algorithm may reduce the number of calculations to 1/100 of those of modal synthesis.

Figure 13 summarizes the potential for improving simulation efficiency by improving both models and algorithms. It indicates the opportunity for reducing the number of calculations by three orders of magnitude.

5. CONCLUSIONS

Now, computer implementation of transient analysis of aircraft structures provides for accurate response predictions. The dynamicist can hope to determine the accuracy of his particular simulation only by "heroic" efforts. Steps he may make to satisfy his desire for efficient analysis are heuristic.

Thus, desirable new technology includes a validated comprehensive set of simulation accuracy and relative efficiency measures. Using these measures to identify research opportunities will lead naturally to better models and data processing algorithms.

The ultimate benefit of accuracy measures will be that dynamicists will have the data they need to more fully understand and interpret the computer's time histories. The ultimate benefit of efficiency measures will be exploitation of the potential to reduce the number of calculations of transient analysis by one to three orders of magnitude. (Fig. 14).

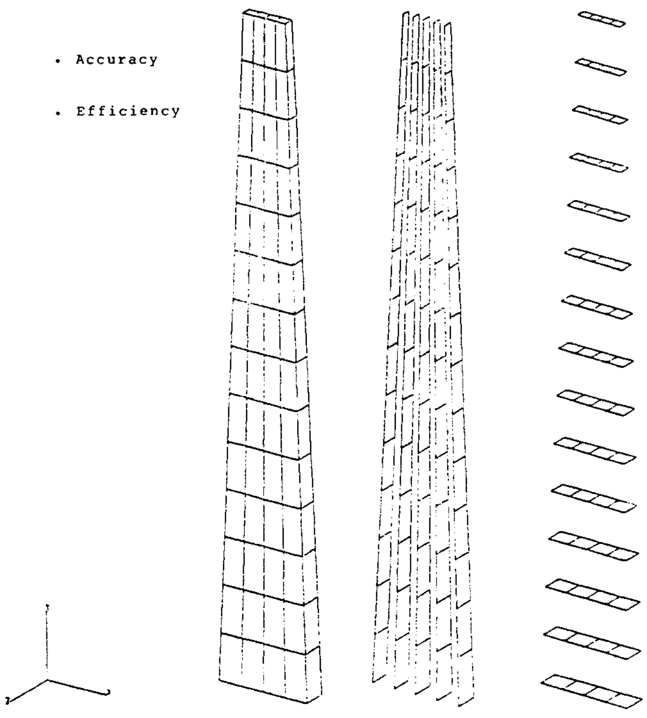


Figure 1. Typical structure: NASA Ames design of oblique wing.

FORM EQUATIONS OF MOTION

- SELECT ELEMENT MODELS
- GENERATE ELEMENT MATRICES
- DESIGN, REDESIGN MESHES
- ASSEMBLE EQUATION COEFFICIENTS

INTEGRATE EQUATIONS OVER TIME

- SELECT BASIC VECTORS
- SELECT INTEGRATION RULES
- REDUCE EQUATIONS
- EVOLVE TIME HISTORIES

EVALUATE RESPONSE DATA

- FIND PEAK DISPLACEMENTS
- EVALUATE STRESSES
- SELECT STRESS FORMULAS
- FIND PEAK STRESSES

INTERPRET SIMULATION RESULTS

- VALIDATE MODELS, PROCESSES
- QUALIFY ANALYSIS RESULTS
- VERIFY PRODUCTION ANALYSIS
- RELATE TO THE PHYSICAL SYSTEM

Figure 2. Tasks of simulation.

FORM EQUATIONS OF MOTION

- Library of element models
- Modeling experience
- Assembly process
- Data generators
- Grid generators and smoothers
- Geometry plotters
- Element model generators

INTEGRATE EQUATIONS OVER TIME

- Modal, Guyan, and Ritz vector technology
- Sparse matrix operations
- Experience in transient analysis
- Duhamel, explicit, implicit, and quasi-static integrators

EVALUATE RESPONSE DATA

- Interpolation logic
- Displacement differentiation, D'Alembert, inertia relief methods
- Peak stress logic

INTERPRET SIMULATION RESULTS

- Element validation tests
- Process validation tests
- Error measures
- Sensitivity analysis
- Tabulations and response plots
- Data retrieval logic

Figure 3. Supporting simulation technology.

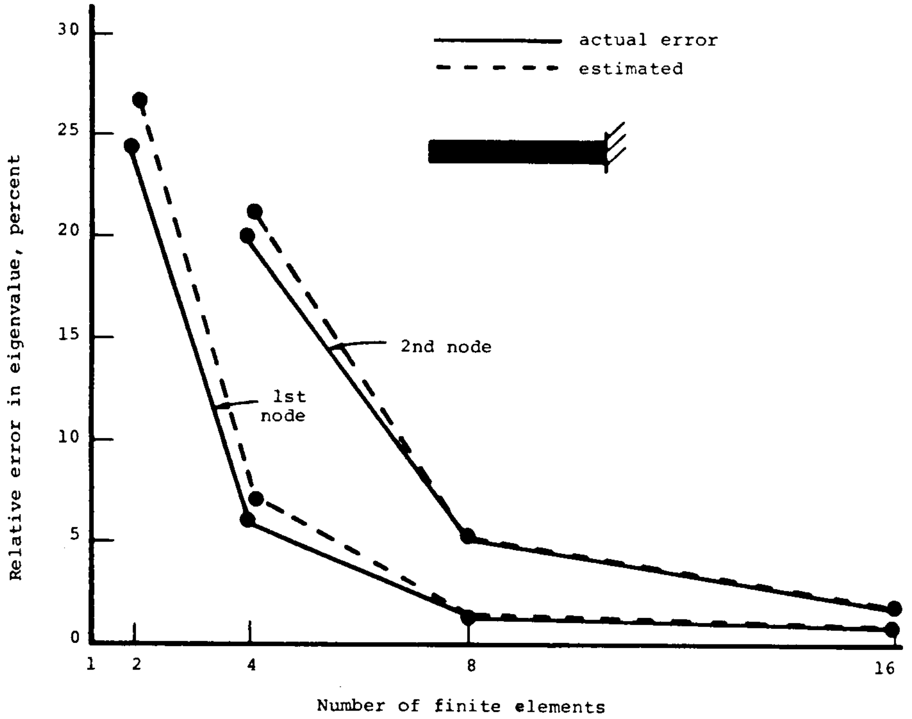


Figure 4. Spatial discretization errors in eigenvalues.

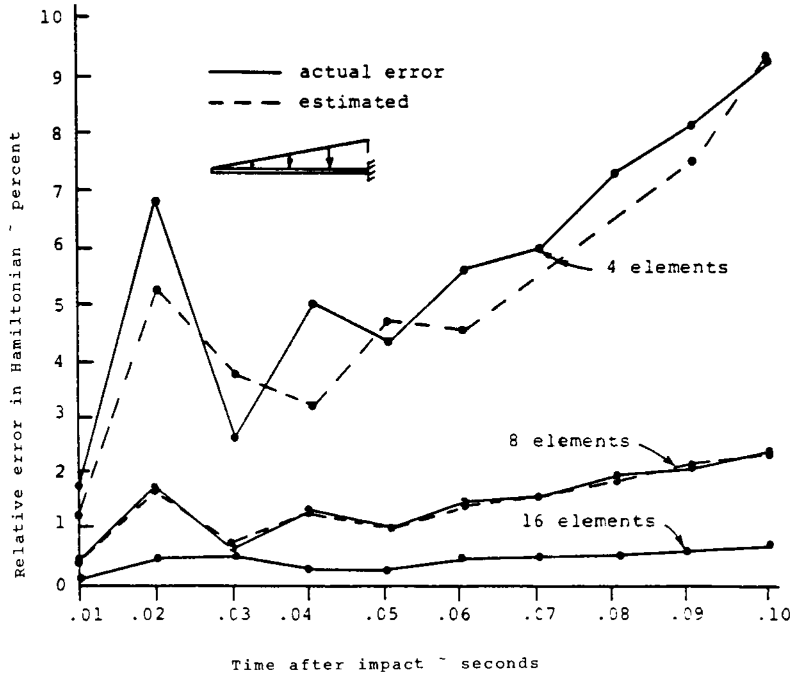
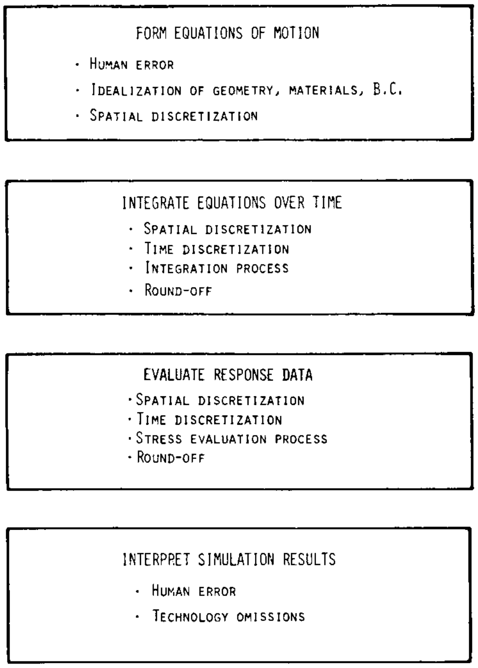


Figure 5. Transient analysis discretization errors.



FORM EQUATIONS OF MOTION

- Human error
- Idealization of geometry, materials, B.C.
- Spatial discretization

INTEGRATE EQUATIONS OVER TIME

- Spatial discretization
- Time discretization
- Integration process
- Round-off

EVALUATE RESPONSE DATA

- Spatial discretization
- Time discretization
- Stress evaluation process
- Round-off

INTERPPET SIMULATION RESULTS

- Human error
- Technology omissions

Figure 6. Sources of inaccuracy in transient analysis.

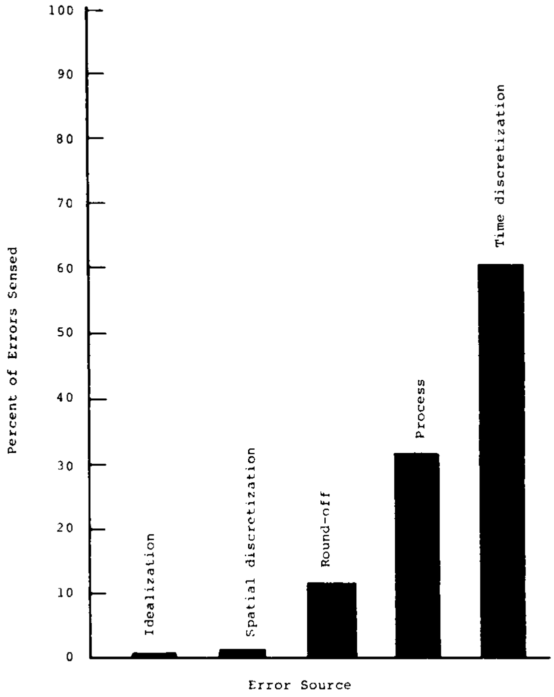
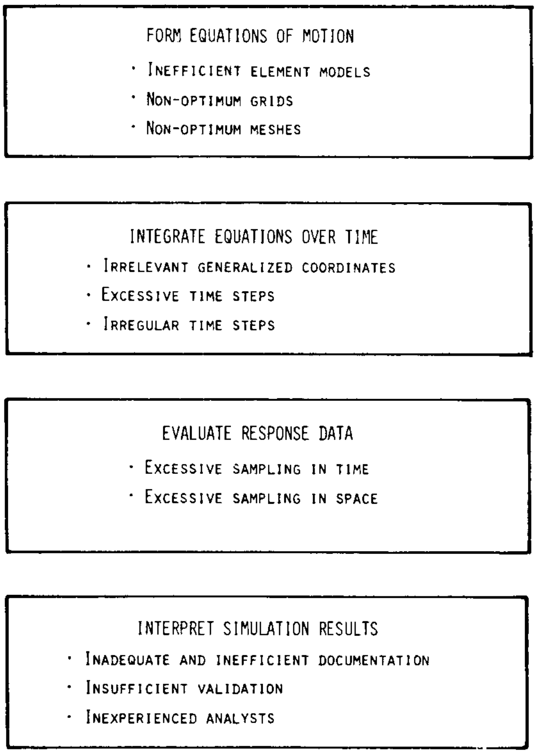


Figure 7. Control of inaccuracies.



FORM EQUATIONS OF MOTION

- INEFFICIENT ELEMENT MODELS
- NON-OPTIMUM GRIDS
- NON-OPTIMUM MESHES

INTEGRATE EQUATIONS OVER TIME

- IRRELEVANT GENERALIZED COORDINATES
- EXCESSIVE TIME STEPS
- IRREGULAR TIME STEPS

EVALUATE RESPONSE DATA

- EXCESSIVE SAMPLING IN TIME
- EXCESSIVE SAMPLING IN SPACE

INTERPRET SIMULATION RESULTS

- INADEQUATE AND INEFFICIENT DOCUMENTATION
- INSUFFICIENT VALIDATION
- INEXPERIENCED ANALYSTS

Figure 8. Sources of inefficiency.

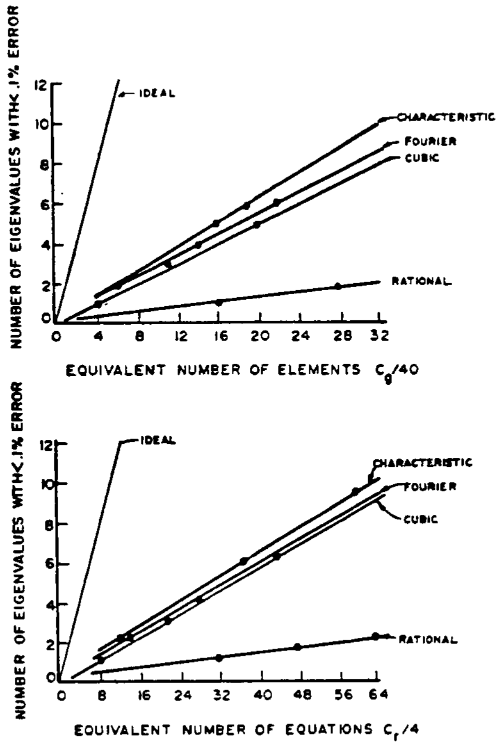


Figure 9. Efficiency of beam element models.

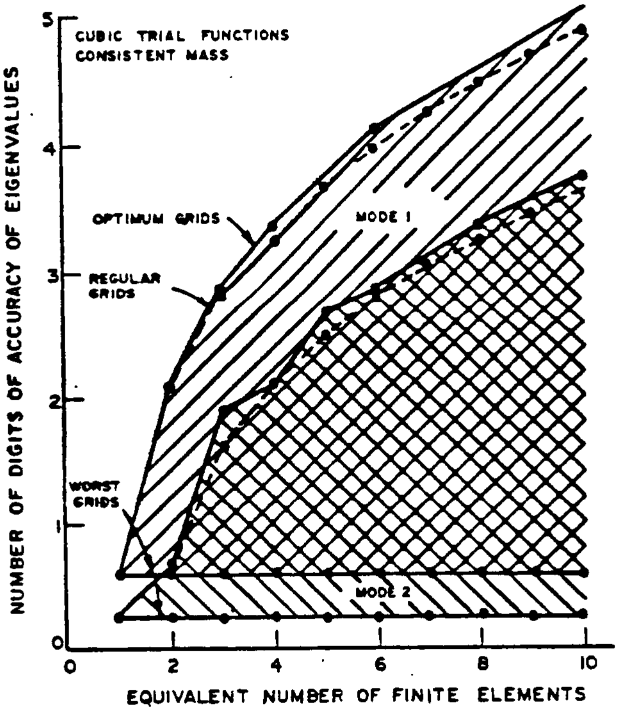


Figure 10. Efficiency of grid designs.

ANALYSIS CLASS (1)	BEST INTEGRATION ALGORITHM	NO. OF CALCS. (2)	BEST/WORST (3)
H.B.	Central or Newmark	Nb^2	N
L.B.	Wilson Ritz	16Nb	b
H.E.	Newmark	$4Nb\frac{t_p}{\Delta t_i}$	N/b
L.E.	Modal Synthesis	$8N\frac{t_p}{\Delta t_i}$	$N\frac{t_p}{\Delta t_i}$

(1)

H = high frequency response important; L = low

B = brief period of integration; E = extensive period

(2)

N = no. of equations of motion

b = semi-band width of integration operator matrix

t_p = period of integration

Δt_i = time step required

(3)

Comparing Central Differences, Newmark Beta, Modal Synthesis, and Wilson's Ritz Vectors methods.

Figure 11. Efficient time integration-linear.

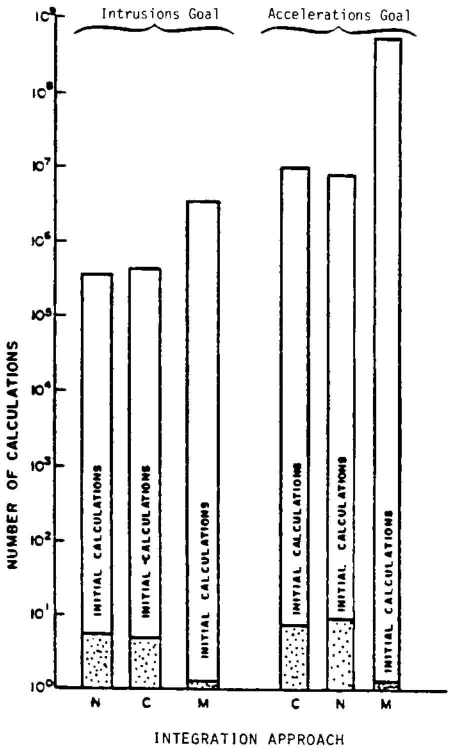


Figure 12. Calculations for nonlinear analysis.

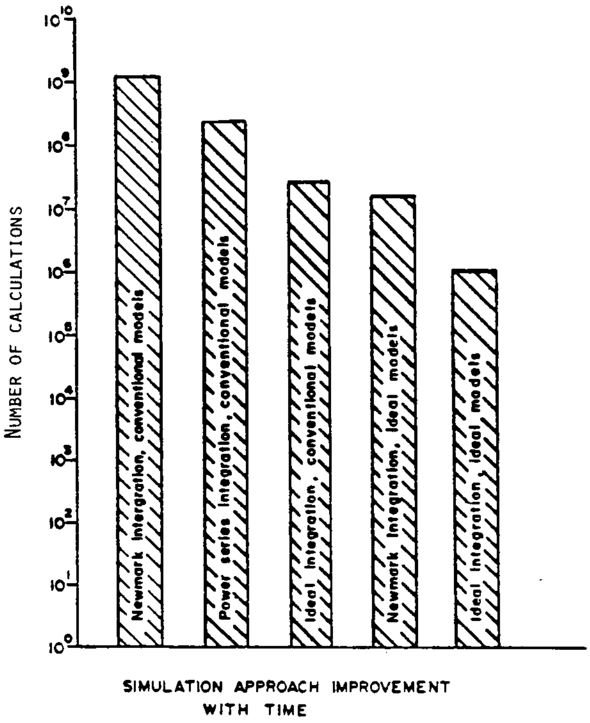


Figure 13. Improvement of simulation efficiency.

STATE-OF-THE-ART

- CAPABILITY FOR ACCURACY EXISTS
- ACCURACY NOT WELL QUANTIFIED
- EFFICIENCY EMPIRICAL

PROSPECTIVE IMPROVEMENTS

- COMPREHENSIVE SET OF ACCURACY SENSORS
- A SET OF EFFICIENCY SENSORS

BENEFITS

- SELF-QUALIFIED TRANSIENT ANALYSIS
- MUCH IMPROVED EFFICIENCY

Figure 14. Conclusions.

EXPLICIT, IMPLICIT, AND HYBRID METHODS

T. Belytschko
Northwestern University
Evanston, Illinois

Time integration methods can be separated into two groups: explicit and implicit. Roughly speaking, methods which do not involve the solution of any algebraic equations are called explicit, whereas those that require the solution of equations are called implicit.

The relative advantages and disadvantages of explicit and implicit methods are summarized in Fig. 1. It is interesting to observe that the positive attributes of these two methods form complementary sets, so that if the positive attributes of the two methods can be combined into a single method, a truly powerful method would be achieved.

An important point which is brought out in Fig. 1 is that whereas implicit methods are unconditionally stable for linear problems, stability does not imply accuracy and in fact the stability of implicit methods has often misled structural analysts into using time steps which yield very poor accuracy. Furthermore, no current time integration will undoubtedly be an important topic for future research.

Relative merits of explicit and implicit integration methods

Explicit

+ very simple and trouble free algorithm, complex phenomena easily included

+ accuracy is assured if Δt stable for large systems

+ no stiffness matrix necessary - saves storage

- conditionally stable, small Δt

Implicit

+ unconditionally stable, large Δt

- complex algorithm with low reliability in nonlinear situations

- accuracy can deteriorate

- Newton form has large core storage requirements

Figure 1

The major trend of the past decade of research on time integration procedures has been hybridization methods so as to take advantage of the complementary nature of the positive attributes of explicit and implicit integration. The types of hybridization are indicated in Fig. 2. References for these methods are as follows: partitioning [1-7], operating splitting methods [8-11], semi-implicit methods [9-12]. It should be noted that the distinction between semi-implicit methods and operator splitting methods is rather fuzzy; both groups of methods try to achieve unconditional stability through some modification of the evolution operator which either completely obviates the need for solving any equations or reduces the size of the system to be solved.

Objective of current research in time integration:

to exploit the advantages of implicit and explicit methods through hybridization (advantages of the two methods are complementary!)

directions:

partitioning: different operators on different parts of the mesh

semi-implicit methods: unconditionally stable methods that require no solution of equations or smaller systems

operator splitting methods: split $\underset{\sim}{A}$ to simplify solution - similar to semi-implicit

Figure 2

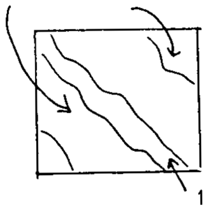
The major shortcoming of operator splitting methods has been the rapid deterioration of their accuracy with increasing time step. For example, if we consider the Trujillo semi-implicit method, which is illustrated in Fig. 3, we find that as the Courant number increases the accuracy diminishes dramatically. In Reference [10] it is shown that the phase velocity in a one-dimensional mesh in the Trujillo method is such that the shorter waves essentially only advance one mesh length during a time step; thus, the effect of the semi-implicit integrator, as shown in Fig. 4, is to retard wave velocity so severely that regardless of the size of the time step a quasi-Courant condition applies in that the numerical waves only traverse a single element in a time step. This distortional characteristic of semi-implicit methods has also been noted in the rigid-body modes by Park and Housner [12]. In Reference [12] several techniques for improving the accuracy of semi-implicit methods were developed, but we have not had time to check their effects independently.

TRUJILLO SEMI-IMPLICIT

(ref. 9)

$$(\underset{\sim}{M} + \Delta t\ \underset{\sim}{K})\ \underset{\sim}{u}^{n+1} = \underset{\sim}{M}\ \underset{\sim}{u}^{n} + \Delta t\ \underset{\sim}{s}^{n+1}$$

let $\underset{\sim}{K} = \underset{\sim}{K}_L + \underset{\sim}{K}_u$



$\frac{1}{2}$ to $\underset{\sim}{K}_L$ and $\underset{\sim}{K}_u$

$$(\underset{\sim}{M} + \Delta t\ \underset{\sim}{K}_L)\ \underset{\sim}{u}^{n+1/2} = (\underset{\sim}{M} - \Delta t\ \underset{\sim}{K}_u)\ \underset{\sim}{u}^{n} + \Delta t\ \underset{\sim}{s}^{n+1}$$

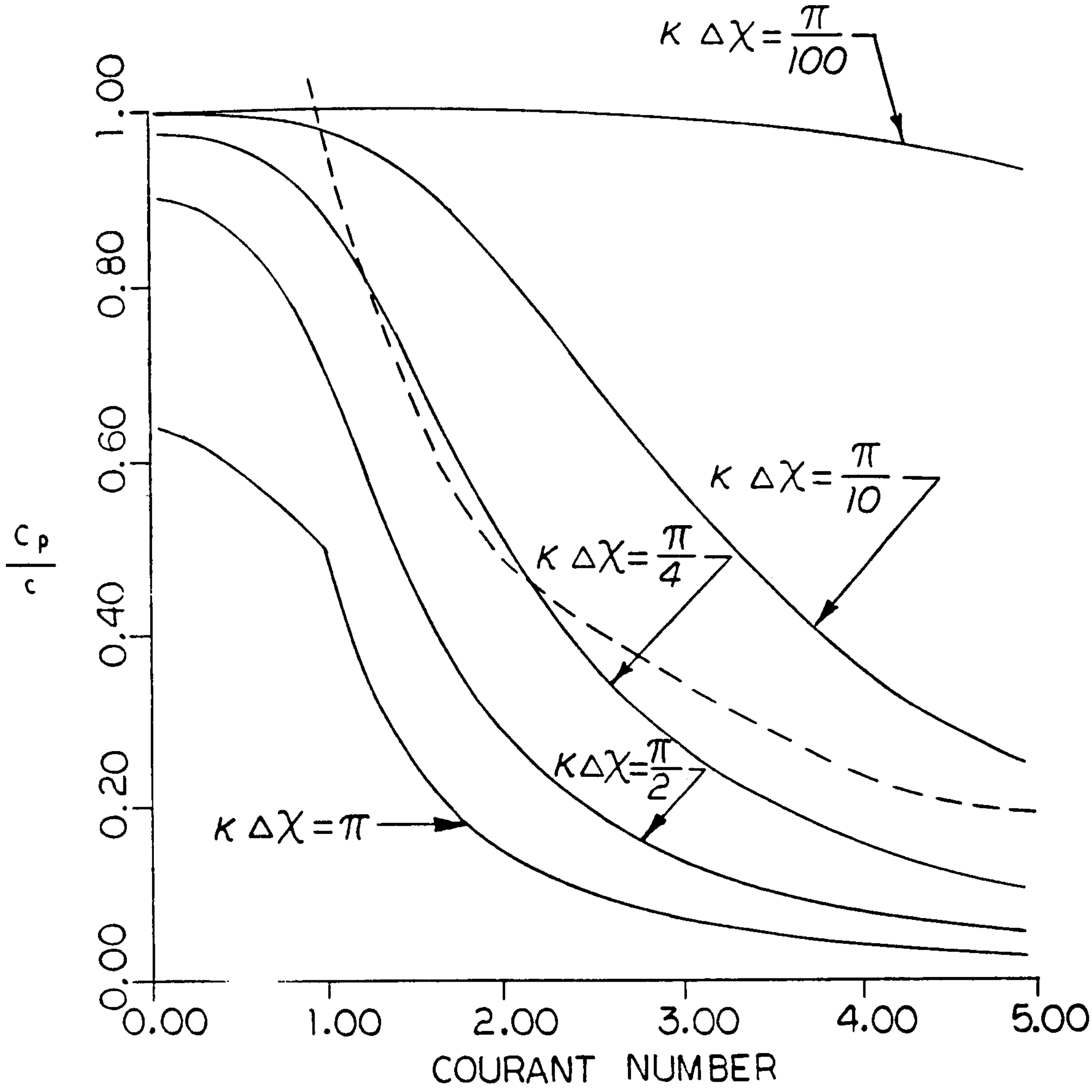
$$(\underset{\sim}{M} + \Delta t\ \underset{\sim}{K}_u)\ \underset{\sim}{u}^{n+1} = (\underset{\sim}{M} - \Delta t\ \underset{\sim}{K}_u)\ \underset{\sim}{u}^{n+1/2} + \Delta t\ \underset{\sim}{s}^{n+1}$$

similar to 2 passes of Gauss - Seidel

unconditionally stable

accuracy?

Figure 3



Phase error as a function of Courant number, r, for various wave numbers in the Trujillo method; the Courant line is dashed and corresponds to $c_p\Delta t=\Delta x$.

Figure 4

Avoidance of equation solution and unconditional stability can be achieved by rational Runge-Kutta methods [13], see Fig. 5. Again, the accuracy of these methods deteriorates rather quickly when the time step is much larger than the stability limit for the explicit methods. These methods seem to be most suited to parabolic systems. For structural mechanics, which involves a combination of hyperbolic and parabolic behavior, their lack of accuracy is generally unacceptable.

Rational Runge Kutta

$$\underset{\sim}{M}\,\underset{\sim}{v}_1 + \underset{\sim}{K}\,\underset{\sim}{\theta}_n = \underset{\sim}{f}_{n+1}$$

$$\underset{\sim}{M}\,\underset{\sim}{v}_2 + \underset{\sim}{K}(\underset{\sim}{\theta}_n + c_2\,\Delta t\,\underset{\sim}{v}_1) = \underset{\sim}{f}_{n+1}$$

$$\underset{\sim}{b} = b_1\,\underset{\sim}{v}_1 + b_2\,\underset{\sim}{v}_2$$

$$\underset{\sim}{\theta}_{n+1} = \underset{\sim}{\theta}_n + \Delta t\ \underset{\sim}{e}/(\underset{\sim}{b}^T\ \underset{\sim}{b})$$

where $\underset{\sim}{e} = 2(\underset{\sim}{v}_1^T\,\underset{\sim}{b})\,\underset{\sim}{v}_1 - (\underset{\sim}{v}_1^T\,\underset{\sim}{v}_1)\,\underset{\sim}{b}$

unconditionally stable and second order accurate

if $c = \frac{1}{2}$, $b_1 = 2$, $b_2 = -1$ Hairer 1980 (ref. 13)

no solution of equations if $\underset{\sim}{M}$ diagonal

$\lambda < 0$ if Δt is too large

partitioned Rational Runge Kutta methods, Liu et al.

IJNME, 1581-1597, (1984), 1984 (ref. 14)

Figure 5

A very novel operating splitting method, which exploits the unique features of the finite-element assembly operation, has recently been developed by Hughes and coworkers [15]. This method only required conversion of the element matrices, so while the method does not completely avoid the solution of equations as in semi-implicit methods, the size of equations to be solved is reduced substantially, see Fig. 6. Hughes and coworkers make a very compelling argument that this type of method will prove particularly beneficial in three-dimensional applications.

We have tested an early version of the method in both parabolic systems and elastic-plastic structural mechanics problems. In comparing the method with a conjugate gradient method, we found that the element-by-element and conjugate gradient methods were of comparable speed. When used with large time steps in structural dynamics problems, we were not able to achieve reasonable accuracy unless we made a large number of sweeps during each time step. On the other hand, we found the method to be very useful in crash-type problems in conjunction with explicit techniques. As a deforming structure becomes mostly plastic, it becomes possible to increase the explicit time step quite a bit if the element-by-element procedure is used to stabilize elements which unload into the elastic regime. This would detract somewhat from the accuracy if it occurs in many element. However, in general, phase accuracy is not an overriding concern in crash-type problems, so that the potential of these methods for stabilizing explicit methods is worth investigating. We have not yet tried the later versions of the element-by-element technique which are reported to be substantially more accurate. Reference [16] reports a procedure which reduces the computational effort required in solving the element equations by as much as an order of magnitude.

ELEMENT-BY-ELEMENT OPERATOR SPLIT

Hughes, Levit, Winget ASCE J. Eng. Mech. Div. April 1983 (ref. 8)
Comp. Meth. Appl. Mech. Eng. 1983 (ref. 15)

Ortiz, Pinsky and Taylor (ref. 17)

Recall implicit Eqns.

$$(\utilde{M} + \Delta t\ \utilde{K})\ \utilde{u}^{n+1} = \bar{\utilde{f}} \qquad (1)$$

$$(\utilde{I} + \Delta t\ \utilde{M}^{-1}\ \utilde{K})\ \utilde{u} = \utilde{f} = \utilde{M}^{-1}\ \bar{\utilde{f}} \qquad \text{(superscript dropped)}$$

(2)

Approximation

$$(\utilde{I} + \Delta t\ \utilde{M}^{-1} \sum_e \utilde{K}_e) = \prod_e \underbrace{(\utilde{I} + \Delta t\ \utilde{M}^{-1}\ \utilde{K}_e)}_{\utilde{G}_e}$$

So (2) becomes

$$\prod_e \utilde{G}_e\ \utilde{u} \equiv (\utilde{G}_1\ \utilde{G}_2 \ldots\ldots \utilde{G}_{ne})\ \utilde{u} = \utilde{f}$$

Procedure

$$\utilde{u}_{[0]} = \utilde{f} \qquad\qquad \utilde{G}_e\ \utilde{u}_{[e]} = u_{[e-1]}$$

ONLY ELEMENT MATRICES NEED TO BE INVERTED!

Figure 6

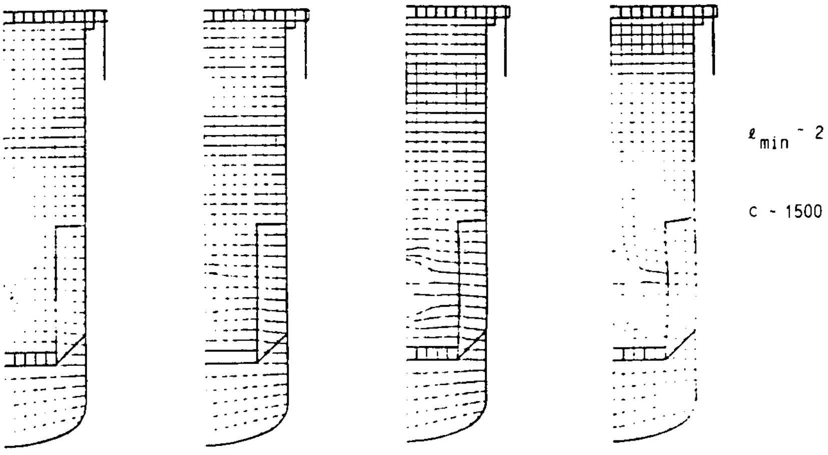
For problems with different time scales such as the space-structure deployment problem, where high-frequency impacts occur in conjunction with low-frequency rigid-body motions, the partitioned methods are quite promising. Partitioned methods are defined as those which employ different time steps or different integrators in different parts of the mesh. During an input, it would be desirable to use different time steps in the vicinity of the impact in solving a large-scale structure problem. By doing this, accuracy could be retained in all parts of the mesh without engendering large expense. The potential of these methods is indicated in Fig. 7.

Δt_{FL} = 0.14 MSEC Δt_{ST} = 0.04 MSEC

RUNNING TIME FOR 60 MSEC SIMULATION

E^1 420 SEC

E^4 - E, E - I 140 SEC IBM 370/195



FOR 60 SEC SEISMIC SIMULATION

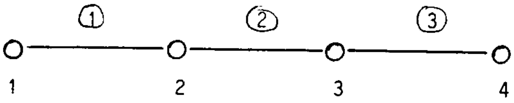
RUNNING TIME E - I: 4.2 X 10^5 SEC

Figure 7

Considerable progress has recently been achieved in mesh partitions with different time steps, see References 5 to 12; 14 to 16, and 18 to 20. Basically, two types of mixed time partitions have been involved: element partitions and nodal partitions. The algorithm for nodal partition is shown in Fig. 8. Nodal partitions appear to provide the best accuracy, but their analysis has been impeded by the fact that the amplification matrix is not symmetric.

Subcycling with Nodal Partition

$$\underset{\sim}{u}^{n+1} = \underset{\sim}{u}^{n} + \Delta t\, \underset{\sim}{M}^{-1} \underbrace{(\underset{\sim}{s}^{n} - \underset{\sim}{K}\, \underset{\sim}{u}^{n})}_{\underset{\sim}{f}^{n}}$$



nodes 1 and 2 with Δt

nodes 3 and 4 with 2Δt

computations in cycle

update u_1, u_2

update $f_{\textcircled{1}}, f_{\textcircled{2}}$

update u_1, u_2

update $f_{\textcircled{i}}, \quad i = 1 \text{ to } 3$

update u_1, u_2, u_3, u_4

update $f_{\textcircled{1}}, f_{\textcircled{2}}$

- amplification matrix is not symmetric

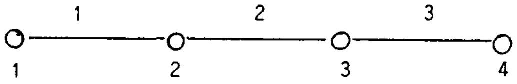
Figure 8

An elemental partition is shown in Fig. 9. Element partitions are associated with symmetric amplification matrices and in Ref. [20] a proof of sufficient conditions or stability has been given for a first-order, linear system with different time steps. The proof applies to both explicit and implicit integrators.

Subcycling with Elemental Partition

$$\underset{\sim}{u}^{n+1} = \underset{\sim}{u}^{n} + \Delta t(\underset{\sim}{M}^{-1})\underbrace{(\underset{\sim}{s}^{n} - \underset{\sim}{K}\,\underset{\sim}{u}^{n})}_{\underset{\sim}{f}^{n}}$$

1st order system i.e. heat conduction diffusion



elements 1 - with Δt

elements 2 & 3 - with $2\Delta t$

computations in cycle

update u_1, u_2

update $f_{\textcircled{1}}$

update u_1, $\underbrace{u_2}$ $\qquad \underbrace{\quad}$ sometimes deleted

update $f_{\textcircled{i}}$ $i = 1$ to 3

update u_1, u_2, u_3, u_4

update $f_{\textcircled{1}}$

amplification matrix eqns are symmetric

Figure 9

Partitioned implicit methods are particularly well-suited to iterative solvers. Whereas for Newton-type solvers, several different triangulations have to be stored for mixed time integration, this is not necessary for iterative solvers. To illustrate the nature of the solutions which can be obtained from these methods the results of the thermal transient problem in Fig. 10 are shown in Figs. 11 and 12. An interesting observation from Fig. 12 is that when the time step ratio is extremely large (32:1 in case 2), stability is maintained but large errors develop. It has become clear that methods of this type must use a smooth transition of time steps from the smallest time step to the largest time step. Thus, an important ingredient in any mixed time integration procedure is a strategy which automatically selects the time steps within the different regimes according to accuracy requirements and provides smooth transitions of time steps between regions where very large time steps can be used and those where very small time steps can be used.

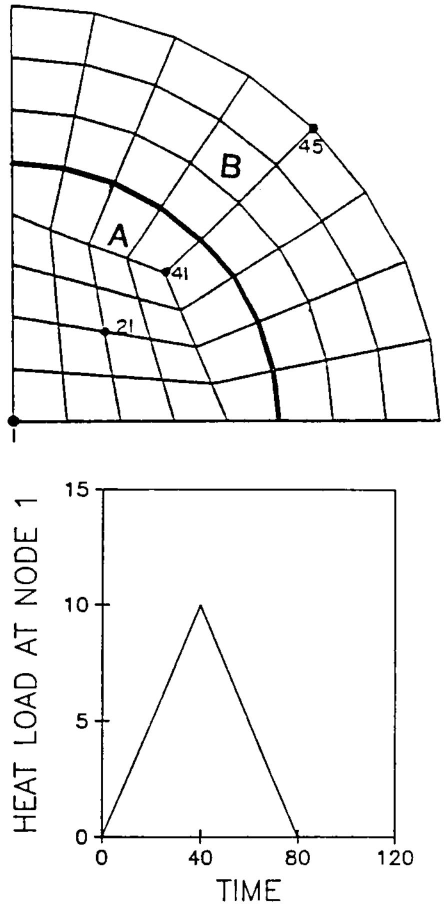
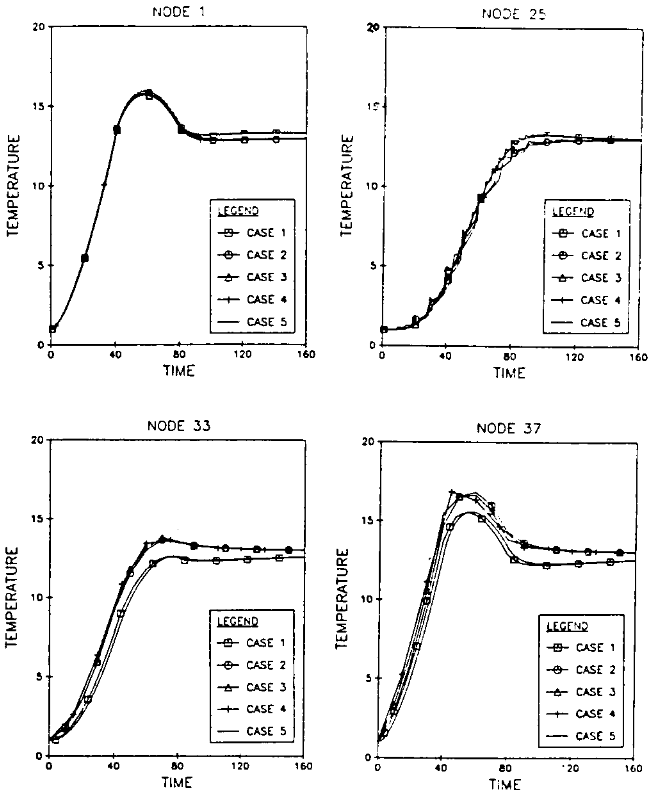


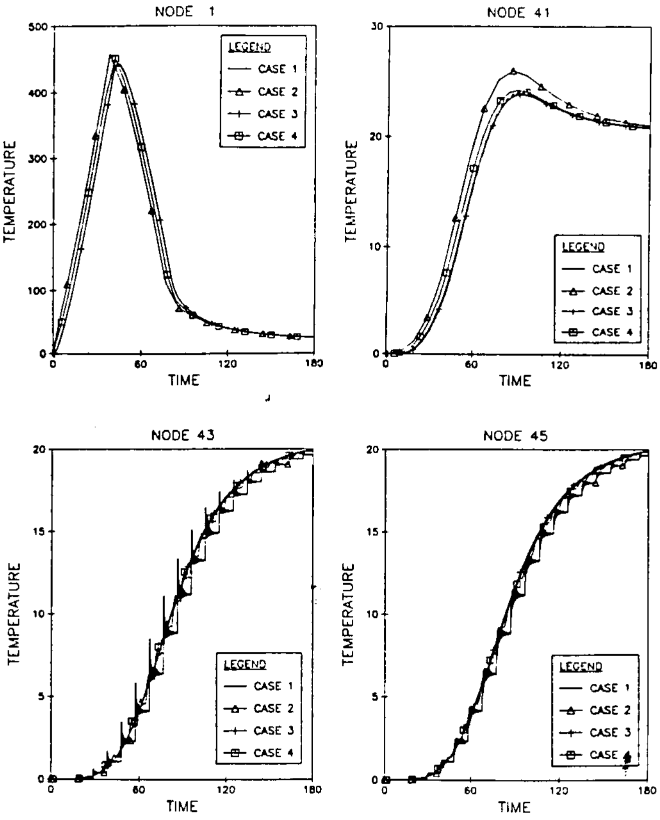
Figure 10



Case 1: E - 4E
2: E - 10I (α = 1)
3: E - 10I (α = ½)
4: 3I - 15I
5: E - E

(From ref. 20)

Figure 11



Case		
Case 1:	E - E,	Δt = 0.3
2:	32I - E,	Δt = 0.3
3:	E - 3I,	Δt = 1.2
4:	5I - I,	Δt = 1.2

(From ref. 20)

Figure 12

The potential of these methods even in two-dimensional problems is quite tremendous, as evidenced by the comparisons shown in Fig. 13. This is a two-dimensional heat conduction problem with a large range of thermal conductivities. As can be seen from the comparison, savings of a factor of 2 to 5 can be achieved even in moderately sized two-dimensional problems. These types of savings have important implications in a computer-aided engineering environment, where the analysis of a new concept must be achieved in a reasonable amount of time if the design process is to be interactive.

These mixed-time integration procedures are in many ways still in their infancy. The applications to nonlinear problems and contact-impact problems will probably require special strategies in order to exploit these methods to their fullest advantage. It would also be desirable to develop stability analyses in the linear regime for second-order systems, such as the equations of motion, and for nodal partitions.

This class of methods, when combined with iterative solvers, would be uniquely suited for parallel architecture computers. In principle, each subdomain with a particular time step could be treated by a different CPU. Data transfer between subdomains would only be necessary for interface data.

Storage and Running Time Comparisons

Storage

Method	I-4E-8E-8I $\Delta t = 1$	I $\Delta t = 1$	I-E-2E-2I $\Delta t = 4$	I $\Delta t = 4$
nonzero terms in $\underset{\sim}{K}$	3288	33771	6089	33771
average semibandwidth	6	69	12	69
solution time CPU-s	90.	516.	70.	139.

Note: • problem is linear;

• 8 x 50 mesh is numbered for large bandwidth to simulate 3D problems.

Figure 13

References

1. T. Belytschko and R. Mullen, "Mesh Partitions of Explicit-Implicit Time Integration," in: Formulations and Computational Algorithms in Finite Element Analysis, K. J. Bathe, J. T. Oden and W. Wunderlich, eds. (MIT Press, Cambridge, MA, 673-690, 1976).

2. T. Belytschko and R. Mullen, "Stability of Explicit-Implicit Mesh Partitions in Time Integration," *International Journal of Numerical Methods in Engineering*, *12* 1575-1586, 1978.

3. T. J. R. Hughes and W. K. Liu, "Implicit-Explicit Finite Element in Transient Analysis," *Journal of Applied Mechanics*, *45*, 371-376, 1978.

4. K. C. Park, "Partitioned Transient Analysis Procedure for Coupled-Field Problems," *Journal of Applied Mechanics*, *47* (2), 370-376, 1980.

5. T. Belytschko, H. J. Yen and R. Mullen, "Mixed Methods for Time Integration," *Computer Methods in Applied Mechanics and Engineering*, *17/18*, 259-275, 1979.

6. J. P. Wright, "Mixed Time Integration Schemes," *Computers and Structures*, *10*, 235-238, 1979.

7. W. K. Liu, "Development of Mixed Time Partition Procedures for Thermal Analysis of Structures," *International Journal of Numerical Methods in Engineering*, 1982.

8. T. J. R. Hughes, I. Levit and J. Winget, "Unconditionally Stable Element-by-Element Implicit Algorithm for Heat Conduction Analysis," *Journal of Engineering Mechanics Division*, ASCE, *109* (2), 576-583. 1983.

9. D. M. Trujillo, "An Unconditionally Stable Explicit Method for Structural Dynamics," *International Journal for Numerical Methods in Engineering*, *11*, 1579-1592, 1977.

10. R. Mullen and T. Belytschko, "An Analysis of an Unconditionally Stable Explicit Method," *Computers and Structures*, *16*, 691-696, 1983.

11. V. K. Saulyev, "Integration of Equations of Parabolic Type by the Method of Nets," Pergamon Press, Elmsford, N.Y., 1964.

12. K. C. Park and J. M., Housner, "Semi-Implicit Transient Analysis Procedures for Structural Dynamics Analysis," *International Journal for Numerical Methods in Engineering*, *18*, 609-622, 1982.

13. E. Hairer, "Unconditionally Stable Explicit Methods for Parabolic Equations," *Numerical Methods*, *35*, 57-68, 1980.

14. W. K. Liu, T. Belytschko and Y. F. Zhang, "Partitioned Rational Runge-Kutta Method for Parabolic systems," *International Journal for Numerical Methods in Engineering*, *20*, 1581-1597, 1984.

15. T. J. R. Hughes, I. Levit and J. Winget, "An Element-by-Element Solution Algorithm for Problems of Structural and Solid Mechanics," Computer Methods in Applied Mechanics and Engineering, 36, 241-254, 1983.

16. T. Belytschko, "An Improved Element-by-Element Semi-Implicit Scheme for Dynamic Problems," Nuclear Engineering and Design, 80 (2), 127-140, 1984.

17. M. Ortiz, P. M. Pinsky, and R. L. Taylor, "Unconditionally Stable Element-by-Element Algorithms for Dynamic Problems," Computer Methods in Applied Mechanics and Engineering, 36, 223, 239, 1983.

18. W. K. Liu, T. Belytschko and Y. F. Zhang, "Implementation and Accuracy of Mixed-Time Explicit-Implicit Methods for Structural Dynamics," Computers and Structures, 19, 521-530, 1984.

19. T. Belytschko, P. Smolinski and W. K. Liu, "Multistepping Implicit-Explicit Procedures in Transient Analysis," in Innovative Methods for Nonlinearl Problems, ed. by W. K. Liu, T. Belytschko, and K. C. Park, Pineridge, Swansea U. K., 135-154, 1984.

20. T. Belytschko, P. Smolinski and W. K. Liu, "Stability of Multi-time Step Partitioned Integrators for First-Order Finite Element Systems, "Computer Methods in Applied Mechanics and Engineering, 1985.

N89-24657

UNCONDITIONALLY STABLE CONCURRENT PROCEDURES FOR TRANSIENT FINITE-ELEMENT ANALYSIS

Michael ORTIZ

Division of Engineering, Brown University

Providence, Rhode Island 02912, USA

Bahram NOUR-OMID

Center for Pure and Applied Mathematics, University of California

Berkeley, California 94720, USA

1. Introduction

The need for more powerful computers has prompted the development of a number of multi-processor machines with multi-tasking capabilities. These are often referred to as *parallel* or *concurrent* processors. In this work we are concerned with the development of time-stepping algorithms for transient finite element analysis which lend themselves to an efficient implementation on parallel computers. This requires the modification of present algorithms to suit the new computing environments. In certain instances, algorithms that have been discarded for applications on sequential processors must be re-examined for possible use on the new parallel machines.

Two essential conditions have to be met for an algorithm to be suitable for concurrent computers:

(1) The algorithm must be such that it divides the problem into sub-tasks which require an approximately equal amount of computational effort.

(2) Each sub-task must be as independent as possible.

The first requirement ensures that all the processors start and end their work almost simultaneously, thereby reducing the idle time. The second condition is formulated with a view to minimizing the transfer of information between processors. In [1] Gentleman pointed out that the time for data communication from one processor to another can be substantial in comparison to computation time.

The element-by-element (E×E) solution procedures [2,3,4] were first proposed to reduce storage requirements on sequential computers. In [5] it was suggested that E×E algorithms are potentially useful for concurrent processing as well. However, a closer examination reveals that although the first aforementioned requirement is met, the data transfer between sub-problems can be substantial. This is mainly due to the fact that E×E methods are based on product algorithms which are ***inherently sequential.***

In this paper a new, fully parallelizable class of solution procedures for transient finite element analysis is outlined. Further details about the method can be found in [6]. The algorithms are such that any part of the structure can be processed independently of the rest over a time step. Thus, for any partition of the structure all the members of the partition can be processed independently and simultaneously, i.e., ***concurrently*** over a time step. The proposed algorithms have the structure of an *explicit scheme.* In particular, ***no global equation solving effort is involved.*** However, the proposed class of algorithmscontains an ***unconditionally stable subclass*** for which the choice of time step size is dictated by accuracy considerations alone. This is a typically implicit-like property.

In sum, concurrent procedures may be regarded as a hybrid of implicit and explicit schemes which exhibits some of the best attributes of both types of methods, such as the unconditional stability of implicit algorithms and the concurrency of explicit schemes. This latter feature renders the proposed algorithms particularly well suited for a parallel environment.

2. A class of unconditionally stable concurrent algorithms

Next we discuss a class of time-stepping algorithms a distinct characteristic of which is that they lend themselves to an efficient implementation on parallel processors. The parallel nature of this class of algorithms owes to the fact that, for any partition of the finite element mesh, each subdomain in the partition can be processed independently of the others over a time step. In particular, one can choose a partition in which the subdomains are the finite elements themselves. In this case, all of the finite elements can be processed concurrently and independently of each other, i.e., ***in parallel.*** It should be emphasized, however, that an element-based partition is just a particular choice among a continuous spectrum of possibilities. In practice, the number of subdomains is limited by hardware considerations such as the number of processors in a parallel computer.

For simplicity, the method is next outlined within the context of linear heat conduction. Further details as well as an extension of the method to the dynamic case can be found in [6]. Upon application of the finite element method as a means of spatial discretization the problem is reduced to a set of semidiscrete equations

$$\mathbf{M}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{f} \tag{1}$$

where $\mathbf{d}$ denotes the nodal temperature array, $\mathbf{M}$ the capacity matrix, $\mathbf{K}$ the conductivity matrix and $\mathbf{f}$ the nodal source vector. In finite element analysis the conductivity and capacity matrices are assembled from element contributions through the assembly operation

$$\mathbf{K} = \sum_{e} \mathbf{K}^{e}, \qquad \mathbf{M} = \sum_{e} \mathbf{M}^{e} \tag{2}$$

where $\mathbf{K}^e$ and $\mathbf{M}^e$ are the element conductivity and capacity matrices, respectively.

The application of the method requires that the structure be first partitioned into subdomains. In multi-processor computers, the number of such subdomains is typically taken to be equal to the number of processors in the machine. It is interesting to note that, unlike the E×E method, the mesh partitions can here be chosen with no concern for the connectivity of the subdomains. This greatly facilitates the definition of mesh partitions. Let $\mathcal{S}$ denote the domain of analysis and $\{\mathcal{S}^a,\ a = 1,\ldots,N\}$ a given partition of the mesh into N subdomains $\mathcal{S}^a$. We shall use the symbols $\mathbf{M}^a$, $\mathbf{K}^a$ and $\mathbf{d}^a$ to denote the mass and stiffness matrices and the local solution array of substructure $\mathcal{S}^a$. Thus, $\mathbf{d}^a$ contains the nodal values of the solution at nodes within $\mathcal{S}^a$ and it fully determines the state of the subdomain. The local matrices $\mathbf{M}^a$ and $\mathbf{K}^a$ are obtained from a partial assembly (2) extended to the elements contained in subdomain a. Furthermore, let $\Gamma^{int} \equiv \bigcup_{a=1}^{N} \partial\mathcal{S}^a - \partial\mathcal{S}$ denote the 'interior boundary' of the partition. In other words, the interior boundary is the union of the parts of the subdomain boundaries which do not lie on the boundary of the overall domain. The restriction of $\mathbf{d}$ to Γ^{int} will be denoted by $\mathbf{d}^{int}$.

With this terminology, the conceptual algorithm can be stated as follows:

(i) Localize the initial conditions $\mathbf{d}_n$ to subdomains $\mathcal{S}^a$ to obtain an extended array $\bar{\mathbf{d}}_n \equiv \{\mathbf{d}_n^1,\ldots,\mathbf{d}_n^N\}$.

(ii) Update local arrays $\{\mathbf{d}_n^a\}$ using an implicit algorithm to integrate the decoupled subproblems

$$\mathbf{M}^a\dot{\mathbf{d}}^a + \mathbf{K}^a\mathbf{d}^a = \mathbf{f}^a \tag{3}$$

Let us denote by $\bar{\mathbf{d}}_{n+1}^*$ the extended predictor so obtained.

(iii) Mass-average $\bar{\mathbf{d}}_{n+1}^*$ at Γ^{int} to obtain $\mathbf{d}_{n+1}^{int}$.

(iv) Integrate again the decoupled subproblems (3) with initial conditions $\mathbf{d}_n^a$ and prescribed all-around boundary conditions $\mathbf{d}_{n+1}^{int}$ to obtain the updated solution array $\mathbf{d}_{n+1}$.

Thus, the basic algorithm involves a double pass through the subdomains in the mesh partition. The sole purpose of the first pass is to determine the updated solution $\mathbf{d}_{n+1}^{int}$ on the interior boundary Γ^{int}. The second pass updates the remaining degrees of freedom for known values of the solution on the subdomain boundaries. It should be noted that in both passes all subdomains can be processed concurrently. For element-by-element mesh partitions one trivially has $\mathbf{d}_{n+1}^{int} \equiv \mathbf{d}_{n+1}$. Under these conditions, the second pass does not alter the solution and can be dropped from the algorithm. On the other extreme, if the structure is not partitioned at all one trivially recovers the implicit schemes.

REMARK 2.1. The choice of a mass-averaging rule is not arbitrary. It can be shown [6] that this is in fact the only choice of averaging rule which renders the algorithm consistent with the global equations of evolution. The mass-averaging rule is implemented as follows. The result of each local update $\mathbf{d}_{n+1}^{*a}$ is first weighted by the local capacity matrix $\mathbf{M}^a$. The resulting local arrays are assembled into a global vector which is then multiplied by $\mathbf{M}^{-1}$. □

REMARK 2.2. The practicality of the method clearly requires the use of a lumped capacity matrix. For most practical purposes, however, this is not a particularly stringent limitation. □

REMARK 2.3. In general, the proposed algorithm can only be expected to be first-order accurate, i.e., $\mathbf{d}_{n+1} = \mathbf{d}(t_n + h) + \mathbf{O}(h^2)$ whenever $\mathbf{d}_n = \mathbf{d}(t_n)$. In [6] it is shown how higher-order algorithms can be derived from the first-order scheme discussed here. □

REMARK 2.4. It should be noted that the updates of the subdomains involve local operations only. In particular, the global stiffness matrix need not be assembled at any time during the integration process, much less factorized. □

REMARK 2.5. A particularly promising feature of the proposed class of algorithms is the fact that they are amenable to a fully *parallel implementation*, whereby all the subdomains in the partition are processed concurrently and independently of each other over a time step. It should be emphasized that the mesh partitions can be defined in a completely arbitrary manner, with no concern for the connectivity of the subdomains. This greatly facilitates the definition of mesh partitions. Another interesting aspect of the algorithm is that exchange of information between the subdomains is only required at the end of a time step. This has the effect of reducing the extent of interprocessor communication to a minimum.

All this is in sharp contrast to other 'semi-implicit' schemes proposed in the past which, although parallelizable to some extent, are inherently sequential, require elaborate schemes to define the mesh partitions and involve interprocessor communications during each time step. □

REMARK 2.6. Clearly, the properties of the proposed concurrent procedures depend on the choice of local update algorithm. It can be shown [6] based on Iron's bounding principle [7] that if the local algorithms are unconditionally stable then resulting concurrent procedure is also ***unconditionally stable.*** In other words, concurrent procedures preserve the stability of the local algorithms utilized to update the subdomains. □

A first numerical example is shown in Fig. 1. The analysis is concerned with linear heat conduction in a bar with prescribed boundary conditions at both ends. The bar was discretized into 100 linear 2-node elements and the resulting mesh partitioned into 4 subdomains each containing 25 elements. The decoupled subproblems were integrated using the backward-Euler algorithm. Fixed time step sizes were utilized throughout the integration process. As may be seen from Fig. 2, the computed results exhibit good accuracy over a wide range of time steps.

3. Accuracy under successive refinements of the partition

The question that naturally arises now is what is the effect on the overall accuracy of the algorithm of successive refinements of the mesh partition. The question is motivated by the observation that the smaller the subdomains in the partition the cheaper is one application of the algorithm. In particular, when element-by-element mesh partitions are utilized the cost of one application of the algorithm is reduced to a minimum. However, numerical experiments inmediately show that increasing the number of subdomains has an adverse influence on the accuracy of the algorithm. This effect is best illustrated by examining the limiting case of element-by-element partitions of the mesh. In this case, the major restriction on the time step size stems from the fact that one application of the algorithm propagates element information to adjacent elements only. This limited flow of information is particularly stringent when analyzing parabolic systems which are far away from equilibrium. In such cases, information needs to be rapidly exhanged between distant sections of the structure. The situation is aggravated by fine meshes for which information has to traverse many elements at the expense of many applications of the algorithm before it is propagated over an appreciable distance. A similar analysis for another class of algorithms has been reported elsewhere [8].

These considerations point to the need of combining element-by-element partitions with a *step-changing technique* to control accuracy. Here the aim is to devise a criterion whereby the time step size is automatically reduced when rapid flow of information is required and increased whenever accuracy permits. A simple strategy is based on Richardson's extrapolation and uses the difference between two solutions $\mathbf{d}_n(h/2)$ and $\mathbf{d}_n(h)$ obtained with step sizes $h/2$ and h, respectively, to estimate the local truncation error as

$$\epsilon_n \approx \| \mathbf{d}_n(h/2) - \mathbf{d}_n(h) \| \tag{4}$$

(see, e.g., [9] where alternative methods are given). Based on this estimate it is possible to determine the extent by which the time step size h has to be reduced or can be increased to satisfy a local trucation error condition

$$\epsilon_n < r \tag{5}$$

for some given tolerance r.

The performance of element-by-element concurrent algorithms can be illustrated by means of the problem stated in Fig. 3. The analysis is concerned with linear heat conduction in a circular region subjected to a sudden temperature rise along the boundary. A mesh of 100 isoparametric 4-node elements was employed. The Crank-Nicolson algorithm (see, e.g., [10]) was utilized for the local updates. The error norm involved in estimate (4) was taken to be $\| \mathbf{d} \| \equiv \left(\frac{1}{n}\sum_{i=1}^{n} d_i^2\right)^{1/2}$ where n denotes the number of degrees of freedom in the model. Fig. 4a shows a comparison between the exact solution and the results obtained for local truncation error tolerances $r = 10^{-4}$ and 10^{-3}. The more stringent tolerance is seen to result in accurate predictions. As larger local truncation errors are allowed, a loss of accuracy is observed which manifests itself as an overly slow relaxation.

The behavior of the step-changing procedure is exhibited in Fig. 4b. It is seen that during the first stages of the relaxation process when the system is far away from thermal equilibrium accuracy demands the use of small time steps. As the system relaxes, the required step size steadily increases. Whereas for explicit integration this growth has to be stopped at the critical time step h_c to avoid numerical instabilities, concurrent algorithms can be used with time steps of any size as accuracy permits. Fig. 4b shows how the critical time step h_c is eventually exceeded without instabilities in the response or any significant loss of accuracy. As a result, the 'average time step', i.e., the duration of the analysis divided by the total number of time steps can be substantially larger for concurrent algorithms than for explicit schemes, which renders the former more economical. In view of this numerical evidence, it should be emphasized that an efficient implementation of the method based on element-by-element mesh partitions within the context of parabolic problems requires that it be combined with a time step-changing technique.

The above numerical results clearly indicate that increasing the number of subdomains in the mesh partition has two opposite effects. On one hand, one application of the algorithm becomes increasingly cheaper. On the other hand, to maintain a given level of accuracy the time step has to be decreased, which adds to the cost of the analysis. The question is which effect dominates and whether using concurrent procedures instead of implicit algorithms is cost effective. That this is so can be illustrated by means of a simple example. Consider the nonlinear 3D dynamic analysis of a cube subdivided into N equal subdomains. The case of implicit integration corresponds to $N = 1$. Numerical tests show that to maintain the same level of accuracy obtained from implicit schemes the time step has to chosen so as to satisfy a Courant condition based on the dimensions of the subdomains. Thus, the time increment has to be decreased as $\mathbf{O}(1/N^{1/3})$ as the number of subdomains increases and consequently the number of steps in the analysis has to be increased as $\mathbf{O}(N^{1/3})$. On the other hand, the number of degrees of freedom per subdomain decreases as $\mathbf{O}(1/N)$ and the bandwith as $\mathbf{O}(1/N^{2/3})$. Therefore, the execution time involved in factorizing a local array decreases as $\mathbf{O}(1/N^{7/3})$. Identifying the cost of one application of the algorithm with that of one local factorization then the total execution time for the analysis goes as $\mathbf{O}(N^{1/3}) \times \mathbf{O}(1/N^{7/3}) \approx \mathbf{O}(1/N^2)$. This shows that concurrent algorithms may be expected to cut down significantly on execution times with respect to implicit algorithms. Similar estimates hold for 2D hyperbolic and 2D and 3D parabolic problems.

4. Summary and conclusions

A new family of algorithms has been outlined which would appear to be particularly well-suited for implementation in a parallel environment. This owes to the fact that for any partition of the mesh each subdomain in the partition can be processed over a time step simultaneously and independently of the rest. The method eliminates the need for assembling and factorizing large global arrays while retaining the unconditional stability properties of the algorithms used at the local level. To critically appraise the proposed methodology, two limiting cases may be considered:

Element-by-element mesh partitions. An appealing feature of element-by-element partitions is that they render the implementation of the method a trivial exercise. Thus, for any finite element code with an explicit algorithm the method can be implemented by merely replacing the usual element stiffness and conductivity matrices by suitably modified ones. Furthermore, this choice of partition has the effect of minimizing storage requirements and arithmetic operations per time step. It is interesting to note that the first order method requires the same number of arithmetic operations per time step as the single pass E×E method. However, for dynamic problems numerical experiments demonstrate the superior accuracy of concurrent algorithms over the E×E method discussed in [3]. For parabolic problems, concurrent algorithms based on element-by-element partitions share the same accuracy limitations as explicit schemes and E×E methods. These limitations arise as a consequence of the limited flow of information per time step allowed by the algorithms. However, as shown above the combination of concurrent algorithms with a step changing technique results in an accurate and reliable procedure which can be significantly more economical that explicit schemes.

Coarse mesh partitions. The use of coarse mesh partitions is a natural choice when implementing the method in concurrent computers. In a parallel environment, the number of subdomains in the partition is dictated by the number of processors in the machine. Remarkably, the numerical evidence presented above shows that the use of coarse mesh partitions is also optimal from the standpoint of both accuracy and cost efficiency. Thus, it would appear that by far the most promising characteristic of the proposed algorithms is their suitability for an efficient and straightforward parallel implementation. By contrast, in this context E×E procedures are cumbersome particularly when applied to structures with complicated topologies. Even for regular meshes the E×E method may not be amenable for a fully parallel implementation. For instance, for a rectangular domain with a regular mesh some degree of parallelism can be obtained from the E×E method as a consequence of the fact that the mesh can be partitioned into four disjoint groups. Then, the elements in each group can be processed concurrently but the groups have to be processed sequentially. Thus, even in this simple case full parallelism cannot be achieved with the E×E method. For arbitrary 2D and 3D topologies a graph coloring algorithm has to be implemented to partition the mesh into disjoint subdomains. This task is by no means trivial. Furthermore, simple bar models can be formulated for which no degree of parallelism at all can be obtained from the E×E method.

In addition, even in the cases where disjoint groups can be easily found, when the processing of a group is completed a set of data pertaining tothe intermediate solution has to be transferred between processors. The time and cost involved in these operations can be substantial [1]. By contrast, the method presented here requires no special partitioning schemes and performs fewer interprocessor communications.

In conclusion, whereas the proposed methodology can be useful in sequential machines as well, it would appear to be most promising as it bears on parallel computation. It should also be emphasized that extensions of the method to nonlinear problems are possible.

References

[1] W. M. Gentleman, Some complexity results for matrix computation on parallel processors, J. Ass. Comput. Mach., Vol. 25 (1978) 112-115.

[2] B.Nour-Ohmidand B. N. Parlett, Element preconditioning, Report PAM-103, Center for Pure and Applied Mathematics, University of California, Berkeley, California, Oct. 1982.

[3] M. Ortiz, P. Pinsky and R. L. Taylor, Unconditionally stable element-by-element algorithms for dynamic problems, Comp. Meth. Appl. Mech. Eng., Vol. 36, No. 2 (1983) 223-239.

[4] T. J. R. Hughes, I. Levit and J. Winget, An element-by-element solution algorithm for problems of structural and solid mechanics, Comp. Meth. Appl. Mech. Eng., Vol. 36, No. 2 (1983) 241-254.

[5] J. M. Winget, Element-by-element solution procedures for nonlinear transient heat conduction analysis, Ph. D. Dissertation, Report CLASSIC-83-05, Center for Large Scale Scientific Computation, Stanford University, Stanford, California, Nov. 1983.

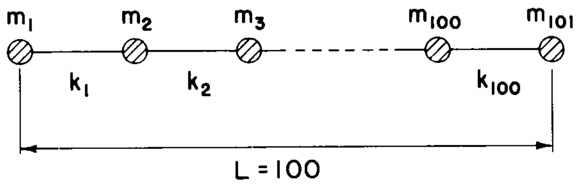
[6] M. Ortiz and B. Nour-Ohmid, Unconditionally stable concurrent procedures for transient finite element analysis, Comp. Meth. Appl. Mech. Engr., (to appear).

[7] B. M. Irons, Applications of a theorem on eigenvalues to finite element problems, (CR/132/70) University of Wales, Department of Civil Engineering, Swansea, 1970.

[8] R. Mullen and T. Belytschko, An analysis of an unconditionally stable explicit method, Computers and Structures, Vol. 16, No. 6 (1983) 691-696.

[9] C. A. Felippa and K. C. Park, Direct time integration methods in nonlinear structural dynamics, Comp. Meth. Appl. Mech. Eng., Vol. 17/18 (1979) 277-313.

[10] T. J. R. Hughes and T. Belytschko, A precis of developments in computational methods for transient analysis, J. Appl. Mech., Vol. 50 (1983) 1033-1041.



MATERIAL PROPERTIES :

$m_1 = m_{101} = 0.5$

$m_2 = m_3 = ---- = m_{100} = 1.0$

$k_1 = k_2 = ---- = k_{100} = 1.0$

BOUNDARY CONDITIONS :

$d_1(t) = 0.0$

$d_{101}(t) = 1.0$

INITIAL CONDITIONS :

$d_2 = d_3 = ---- = d_{100} = 0.0$

Fig. 1. **Definition of test problem: heat conduction in a bar with prescribed temperatures at both ends.**

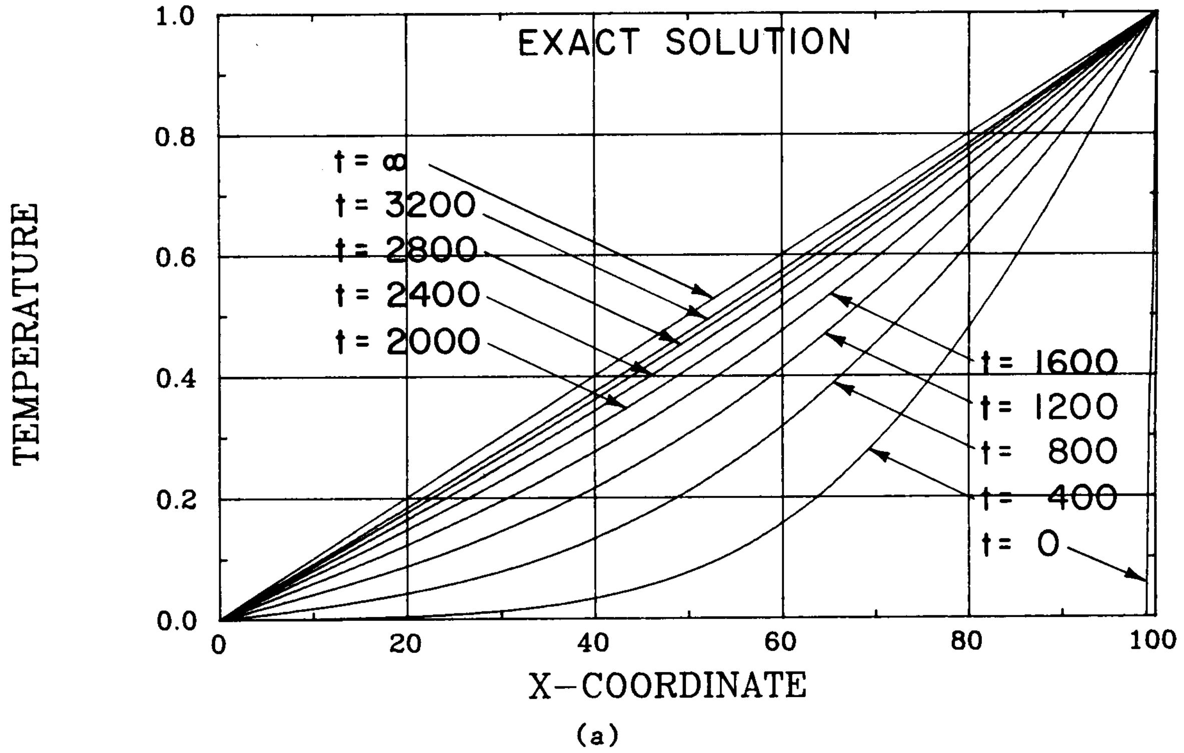
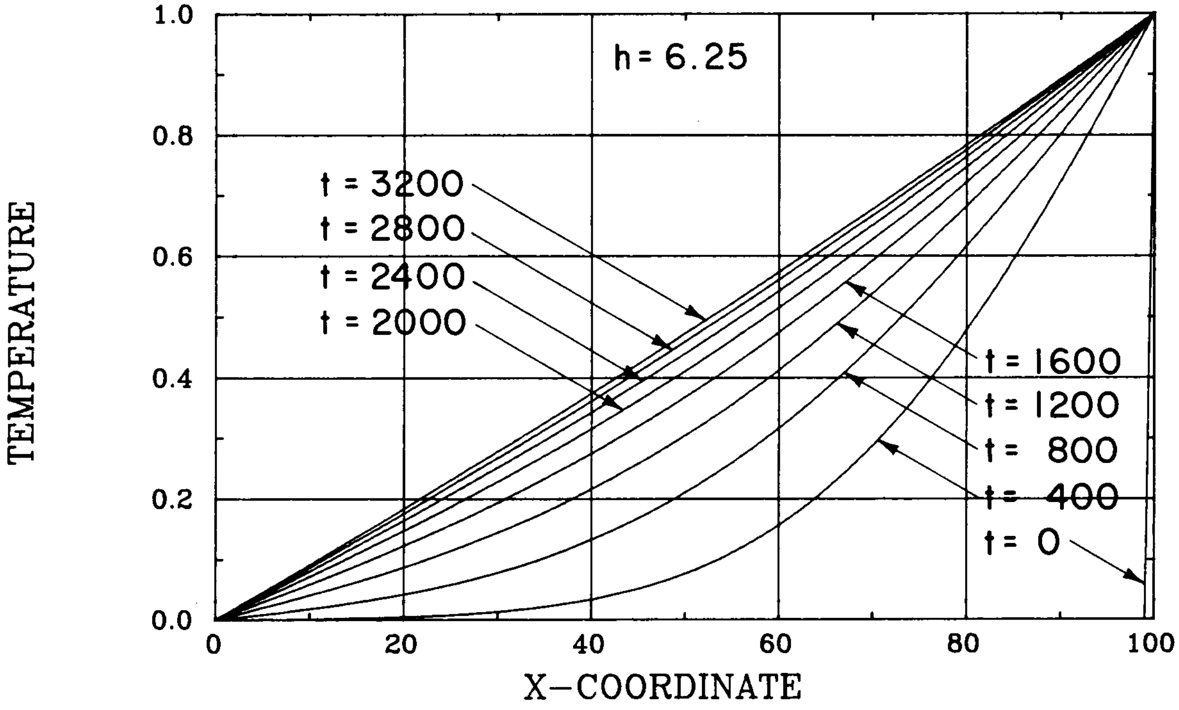
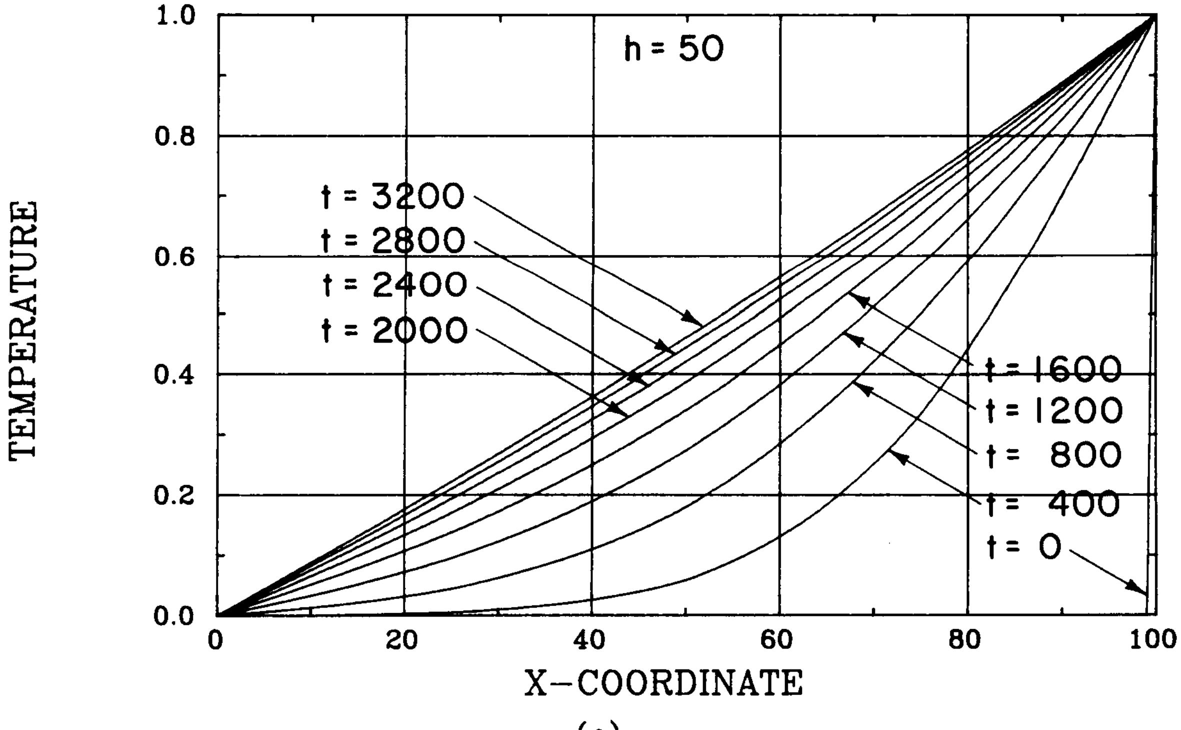


Fig. 2. **Computed results for problem in Fig. 1 using concurrent algorithm based on a partition of the mesh into four subdomains.**



(b)



(c)

Fig. 2. Concluded.

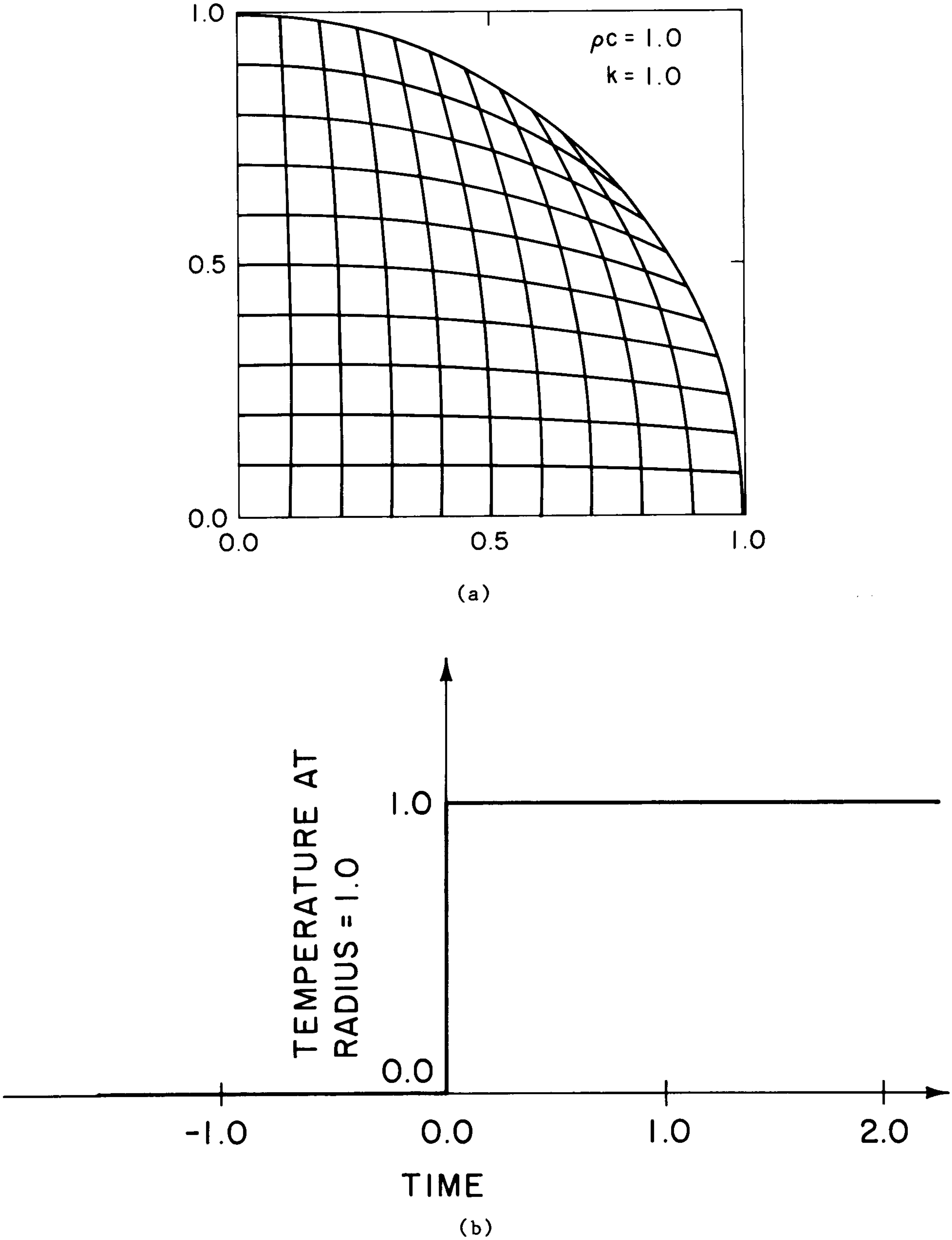
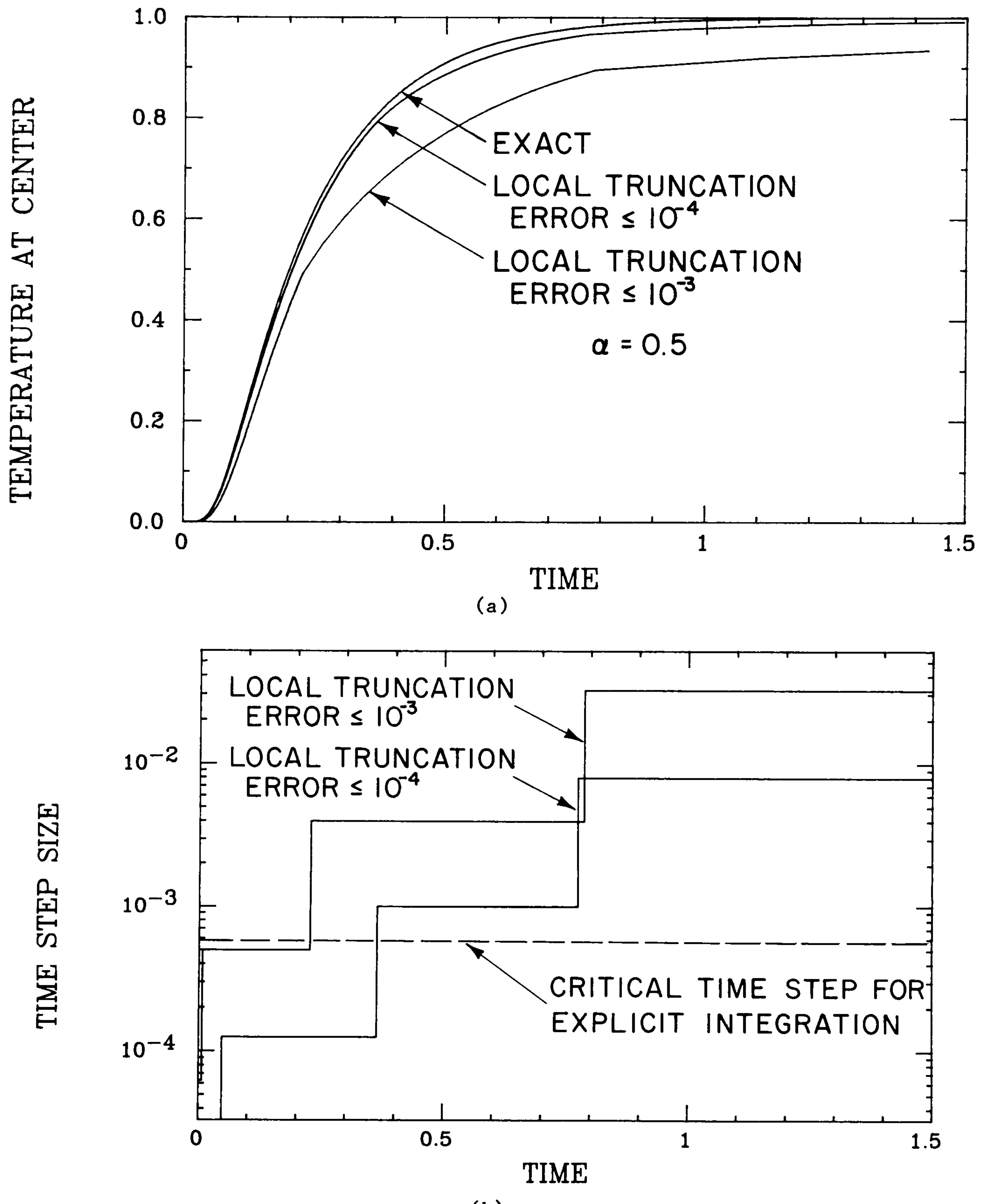


Fig. 3. **Definition of test problem: heat conduction in a circular domain subjected to a sudden rise in temperature along its exterior boundary.**



(a)

(b)

Fig. 4. **Results computed from element-by-element concurrent algorithm with step-changing scheme for problem in Fig. 3. a) Time evolution of temperature at center node. b) Step size variation.**

TRANSIENT ANALYSIS TECHNIQUES IN PERFORMING IMPACT AND CRASH DYNAMIC STUDIES

A. B. Pifko and
R. Winter
Grumman Corporation
Bethpage, NY

Advances in technology are made typically in response to new performance requirements. The area of crash simulation is no exception. Because of the emphasis now being placed on crashworthiness as a design requirement, increasing demands are being made by various organizations to analyze a wide range of complex structures that must perform safely when subjected to severe impact loads, such as those generated in a crash event.

The ultimate goal of crashworthiness design and analysis is to produce vehicles with the ability to reduce the dynamic forces experienced by occupants to specified acceptable levels, while maintaining a survivable envelope around them during a specified crash event.

Figures 1 through 3 show examples of the type of impacts that must be simulated.

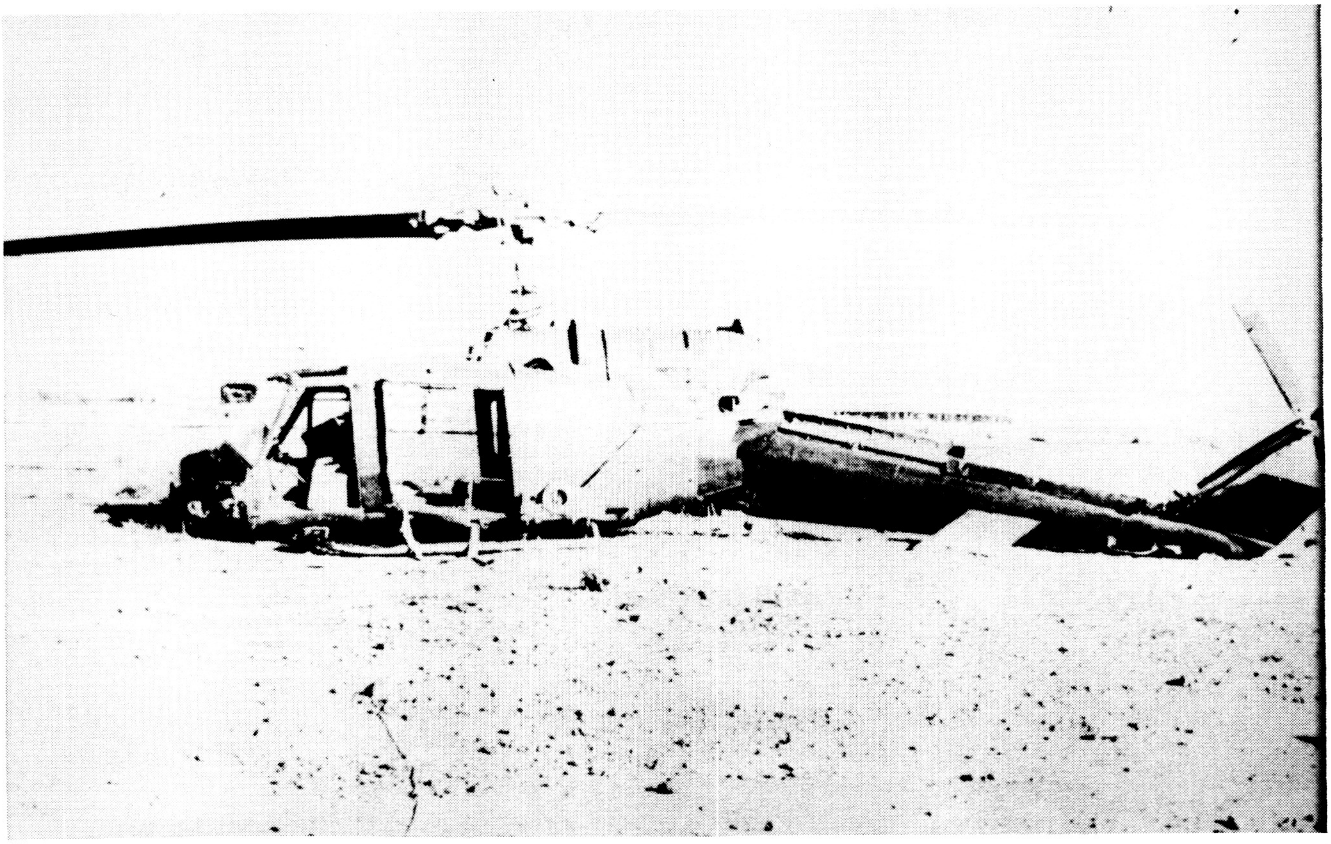


Figure 1. Vertical impact of helicopter.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

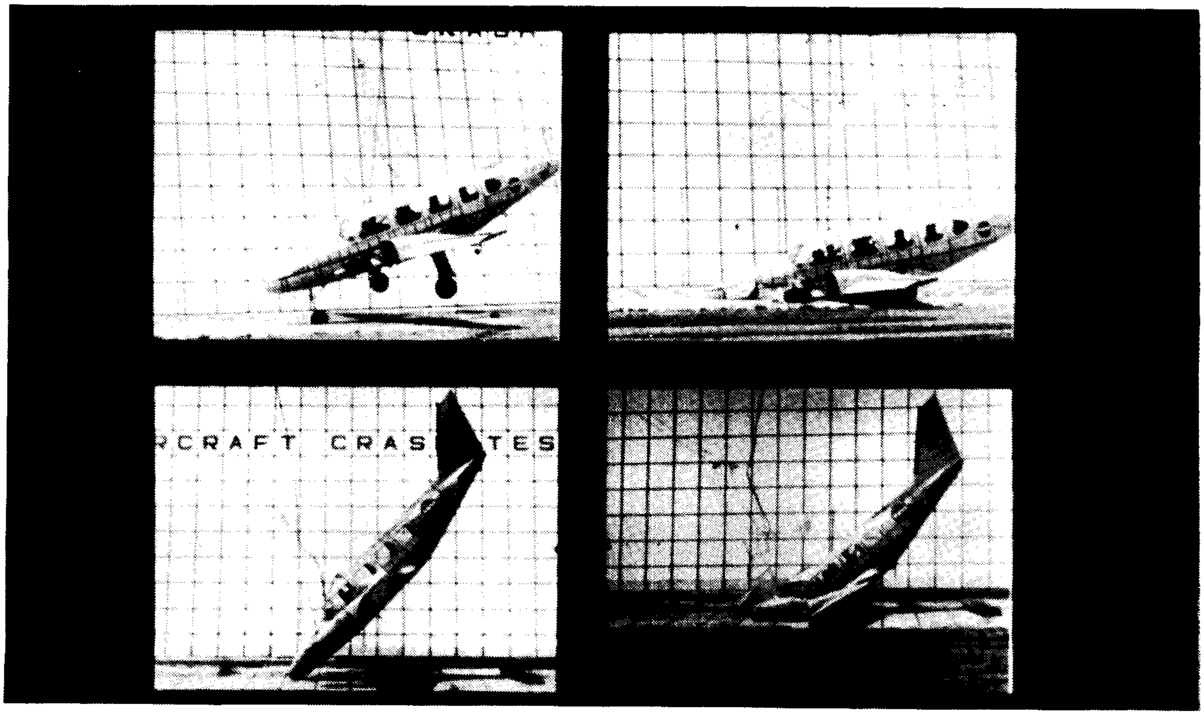


Figure 2. NASA/FAA general-aviation crash dynamics program.

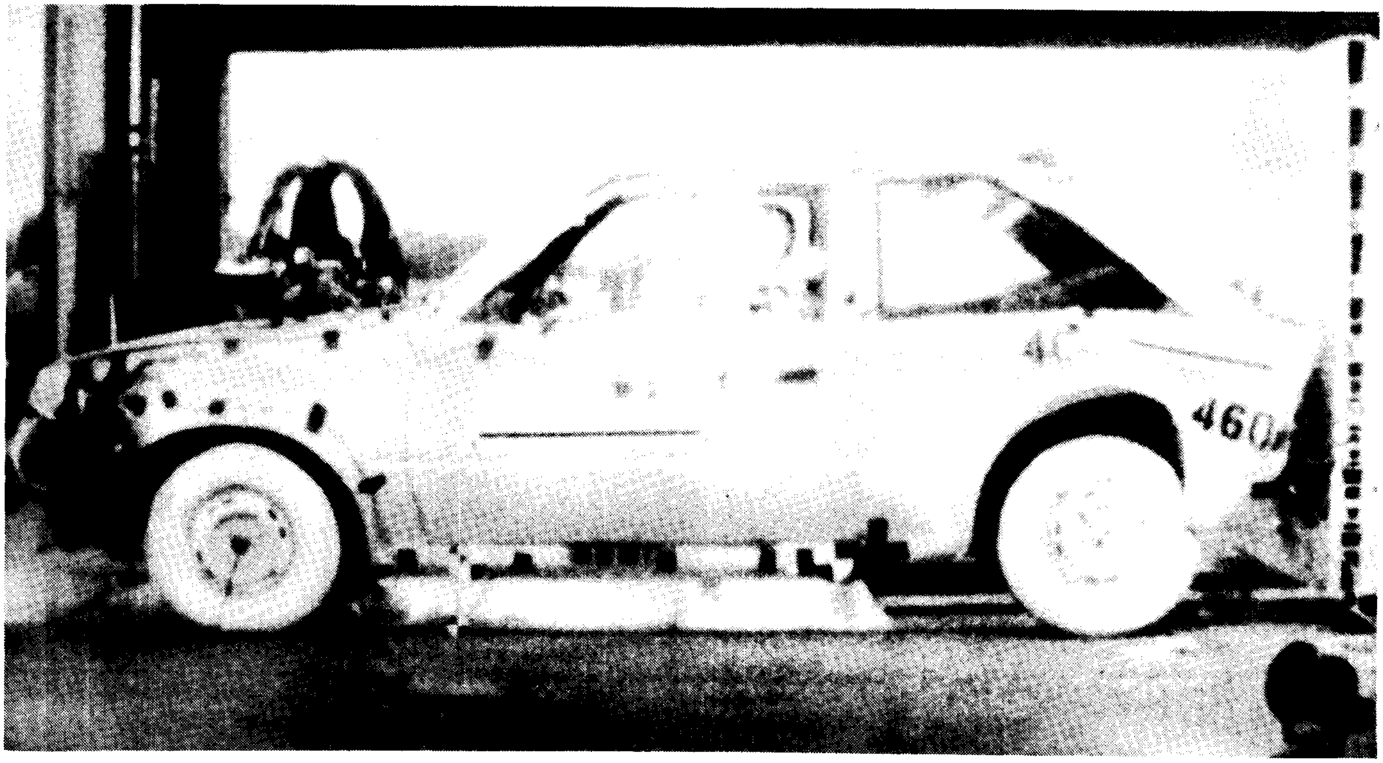


Figure 3. Rear-impact test of automobile.

The requirement for crashworthy vehicles has been a motivating force behind the development of computer programs for use in a vehicle crash simulation. Development of these programs has been the direct result of advances in both structural mechanics and computer sciences. Specifically, advances in finite-element methods, made feasible by rapid developments in computer hardware and software, form the foundation on which these programs were developed. After more than a decade of development, a number of programs are now available and are used for practical analysis and design.

The capability of one such program is reviewed and some experiences gained in the crash evaluation of automobile and aircraft structures are related.

There are a number of requirements that are essential to the simulation of a crash event (fig. 4). Although these requirements involve several areas, the most obvious are:

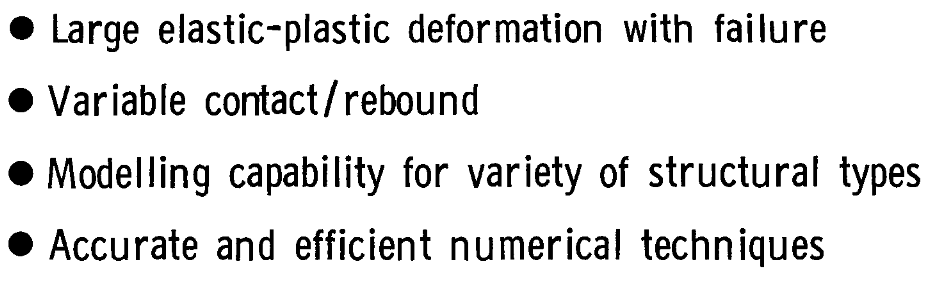
- A theory that treats the large elastic-plastic deformation associated with crushing of structural members including strain-rate effects where applicable

- The techniques for nonlinear boundary conditions required to simulate internal contact/rebound between structural parts or between structure and a barrier or contactor

- A capability to model a variety of structural types, typical of aircraft, and automotive structures

- Accurate and efficient numerical techniques for integrating the nonlinear equations of motion

These requirements include all of the areas that are the subject of current research in computational mechanics. However, methods to treat the essential features of all of these requirements have reached a sufficient level of maturity to be implemented into a code for crash simulation. As such, techniques that account for the essential features of each of the above stated requirements have been incorporated into our DYCAST code.



- Large elastic-plastic deformation with failure
- Variable contact/rebound
- Modelling capability for variety of structural types
- Accurate and efficient numerical techniques

Figure 4. Essential requirements of structure crash simulation.

DYCAST is a nonlinear structural dynamic finite-element computer code that started from the plans system of finite-element programs for static nonlinear structural analysis (fig. 5). It was originally developed for aircraft crash analysis with partial support by NASA Langley.

The equations of motion used in DYCAST are developed within the framework of the finite-element method and are based on the updated Lagrangian formulation for geometric nonlinearity and an incremental plasticity theory for material nonlinearity.

The updated Lagrangian approach is particularly effective for the nonlinear problem associated with crash simulation using beam, membrane, and plate elements. This is because large shape changes due to the progressive crushing and folding of the structure are accounted for by successive updating of the nodal coordinates. The nonlinearities due to the internal loads (for example, the change in stiffness due to the "beam column effect") are included so that compressive forces dominant in a crash event will act through the geometric nonlinearities to reduce the stiffness of the structure.

The following figure outlines the essential features of DYCAST. Our intent in presenting these features is to indicate our view of the necessary minimum requirements for crash analysis.

- Material nonlinearity
    - Incremental plasticity theory
    - Von Mises yield criterion
    - Kinematic hardening
    - Element maximum strain failure criterion
    - Subincremental strategy
- Geometric nonlinearity
    - Updated Lagrangian

Figure 5. DYCAST - Dynamic crash analysis of structures.

The governing matrix equation for the updated Lagrangian formulation is:

$$[M]\ \{\ddot{U}\}_{t+\Delta t} + [K_T + K_G]\ \{\Delta U\}_{t+\Delta t} + \{F\}_t = \{P\}_{t+\Delta t} \qquad (1)$$

Equation 1 is the linearized equation of motion between a known equilibrium state, denoted by t, and an unknown equilibrium state, denoted by t + Δt, incrementally adjacent to it. It explicitly contains terms that reflect the current material state, and nonlinearities from the strain displacement relations.

The quantities in equation 1 are defined as follows: $\{\ddot{U}\}_{t+\Delta t}$, $\{\Delta U\}_{t+\Delta t}$ are the unknown accelerations and displacement increments, $[M]$, $[K_T]$, $[K_G]$ are the mass, tangent stiffness, and initial stress stiffness matrices, respectively, and $\{F\}_t$, $\{P\}_{t+\Delta t}$ are the known internal and external forces at the time denoted by their subscripts.

The matrix $[K_T]$ is a function of the material behavior and therefore explicitly contains the plasticity theory implemented in the code. We have implemented the Prager-Ziegler kinematic hardening theory based on the Von Mises-Hill yield criterion for orthotropic (and isotropic) materials and used an effective stress-strain relation for multiaxial stress states. Postyield behaviors can be either: no strain hardening (perfect plasticity), linear hardening, or nonlinear hardening. Additionally, a multistep subincremental strategy has been employed to ensure that the plastic constitutive equations embodied in $[K_T]$ are never violated.

Assuming continuing and unlimited elastic or plastic deformation in a crash simulation is equivalent to assuming that a structural element will dissipate unlimited energy as it deforms along a particular load-deformation path. Obviously this can overpredict the energy that can be dissipated since actual materials will fail at some maximum deformation. To accommodate this behavior, maximum strain failure criteria have been implemented in our material model. Once these criteria are satisfied at a point, the stiffness and force contributions at that point are deleted. When a specified set of such points in an element has reached its failure strain, the element's stiffness and force contribution to equation (1) is not assembled. Provision has also been made to delete elements manually based on some other failure criterion or on engineering judgment. (Fig. 6.)

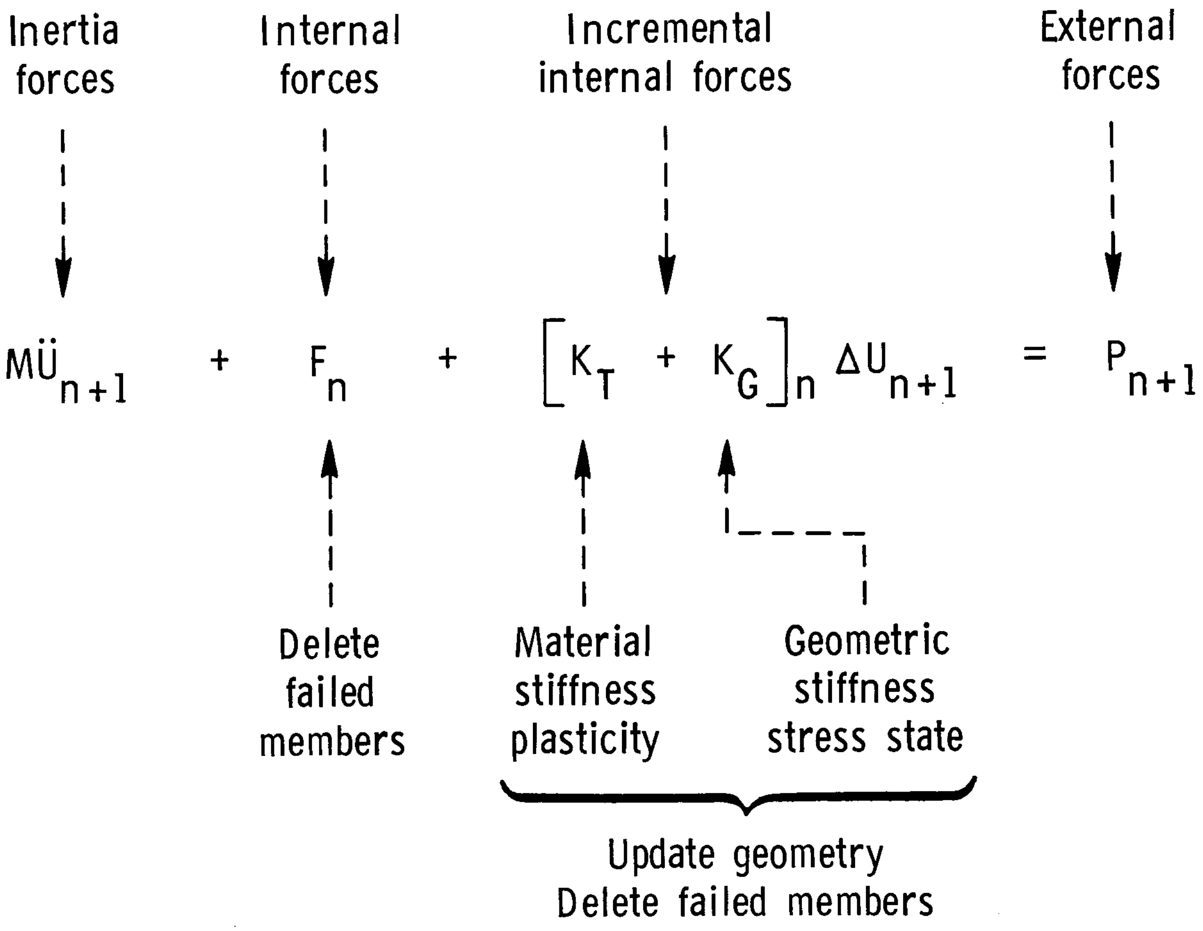

Inertia forces
Internal forces
Incremental internal forces
External forces
$M\ddot{U}_{n+1} + F_n + \left[K_T + K_G\right]_n \Delta U_{n+1} = P_{n+1}$
Delete failed members
Material stiffness plasticity
Geometric stiffness stress state
Update geometry
Delete failed members

Figure 6. Nonlinear dynamics - equation of motion.

At an early stage, it was clear that we should implement a variable time step integrator, i.e., one that enables the time step to be changed at different instants of the response. Such a procedure has obvious advantages over one with a constant step, particularly in complex problems arising in practical application where system nonlinearities and dynamic response are varying continuously throughout the history. Our experience has indicated that this is particularly true for crash simulation.

Variable time step integrators of both the explicit and implicit type have been implemented in DYCAST. These are the explicit Modified Adams and the implicit Newmark-Beta and Wilson-Theta methods. An explicit constant step central difference integrator is also available, as well as static, bifurcation buckling, and free vibration options.

Implementation of implicit integration in DYCAST is as follows: The technique used solves equation (1) at each step subject to the integrator recurrence relations and then performs iterations of the modified Newton type based on an imbalanced force stemming from errors in satisfying the equation of motion.

A variable time step procedure is defined by requiring that the number of iterations in each time step be less than a prescribed value. If this criterion is violated, the time step is halved. Conversely, if the solution converges in one iteration for a prescribed number of consecutive steps the time step is increased. An upper bound for the time step is also specified.

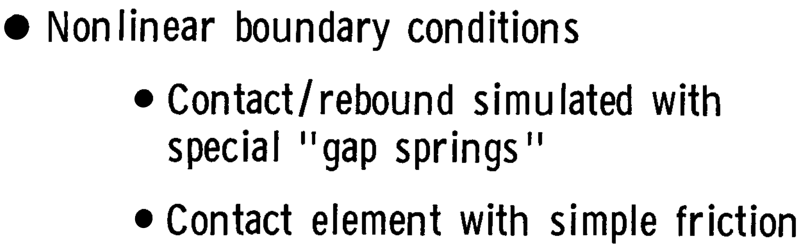
In our initial work, we used the explicit modified Adams integrator exclusively. However, we quickly found that the admissible time step for a nonuniform mesh with beam, plate membrane elements, and nonlinear springs was unreasonably small. Consequently, our current activities are associated almost exclusively with the implicit implementation. Our experience in what we describe as moderate sized problems of 1500 degrees of freedom (DOF) or less has led to a preference of the implicit method in this problem class because solution time per increment is divided almost equally between element level calculations and global solution of a matrix equation. As the number of degrees of freedom increase, the solution of the global matrix equation begins to dominate. Some later figures show examples of some typical calculations. Experience with vector processing (CRAY 1 or CDC CYBER 205) has extended this degree-of-freedom range.

Research is continuing that, hopefully, will address these issues further in such areas as mixed explicit-implicit integration, subcycling, and element-by-element solution strategies that can utilize concurrent processing. (Fig. 7.)

- Implicit
  - Variable time step
  - Modified Newton iteration
  - Newmark -β
- Explicit
  - Modified Adams — variable step
  - Central difference — constant step

Figure 7. Integration of equations of motion.

Nonlinear springs are useful to model the crush behavior of components for which data are available and whose behavior may be too complex to model otherwise (e.g., for energy-absorbing devices, for gap elements with variable contact/rebound, for nonlinear moment-rotation curves of collapsing beams, and for various other nonlinearities). (Fig. 8.)



- Nonlinear boundary conditions
  - Contact/rebound simulated with special "gap springs"
  - Contact element with simple friction

Figure 8. DYCAST - Dynamic crash analysis of structures.

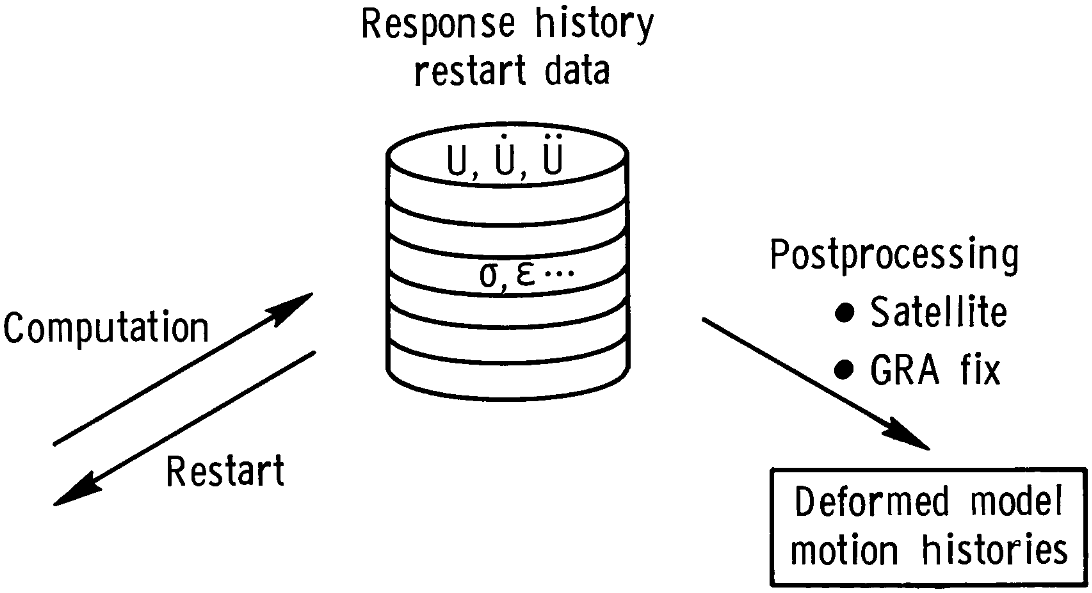
The capabilities outlined represent the essential requirements for a crash simulation. There are other requirements that can be described as operational features, which nevertheless, are essential to the performance of a simulation in an efficient and timely manner. The most important of these is an efficiently designed restart procedure. In keeping with the path dependent nature of nonlinear analysis, this capability enables an analyst to perform a crash analysis in manageable time segments and to examine intermediate results to see if they appear meaningful before deciding if the analysis should be continued.

Adjunct to this is the manner used to display the results. Because the volume of data that is generated can easily overwhelm an analyst, selected summary tables of results along with graphical display are important. Postprocessing graphics include the display of the deformed model at any time and plot histories of displacements, velocity, and acceleration for any nodal degree of freedom.

Access to the restart file by a peripheral program to selectively print additional data is also desirable.

Figures 9 through 12 show necessary process control for restart and some examples of postprocessing graphics.

- Stop, alter, restart, postprocessing



Response history
restart data

$U, \dot{U}, \ddot{U}$

$\sigma, \varepsilon \cdots$

Computation

Restart

Postprocessing
- Satellite
- GRA fix

Deformed model
motion histories

Figure 9. DYCAST - process control.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

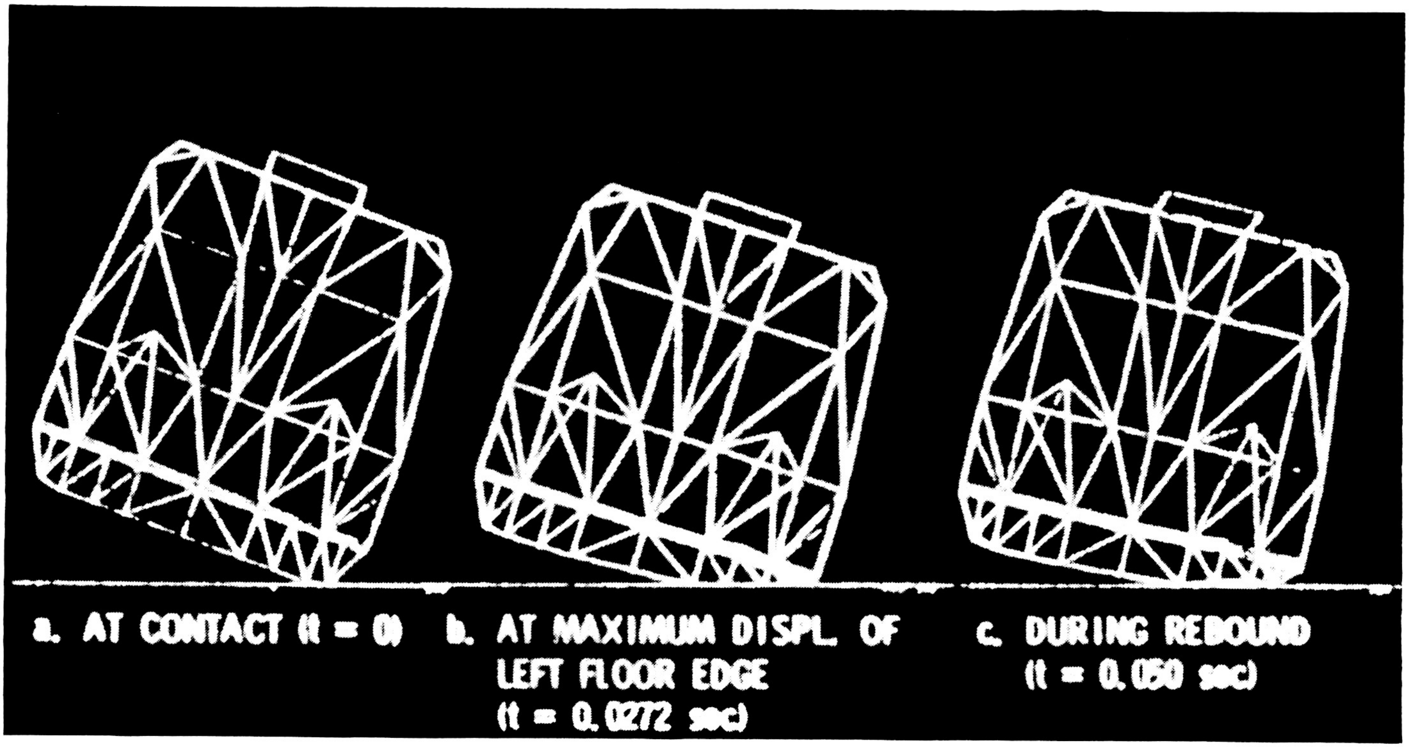

a. AT CONTACT (t = 0)
b. AT MAXIMUM DISPL. OF LEFT FLOOR EDGE (t = 0.0272 sec)
c. DURING REBOUND (t = 0.050 sec)

Figure 10. Deformed DYCAST model roll drop.

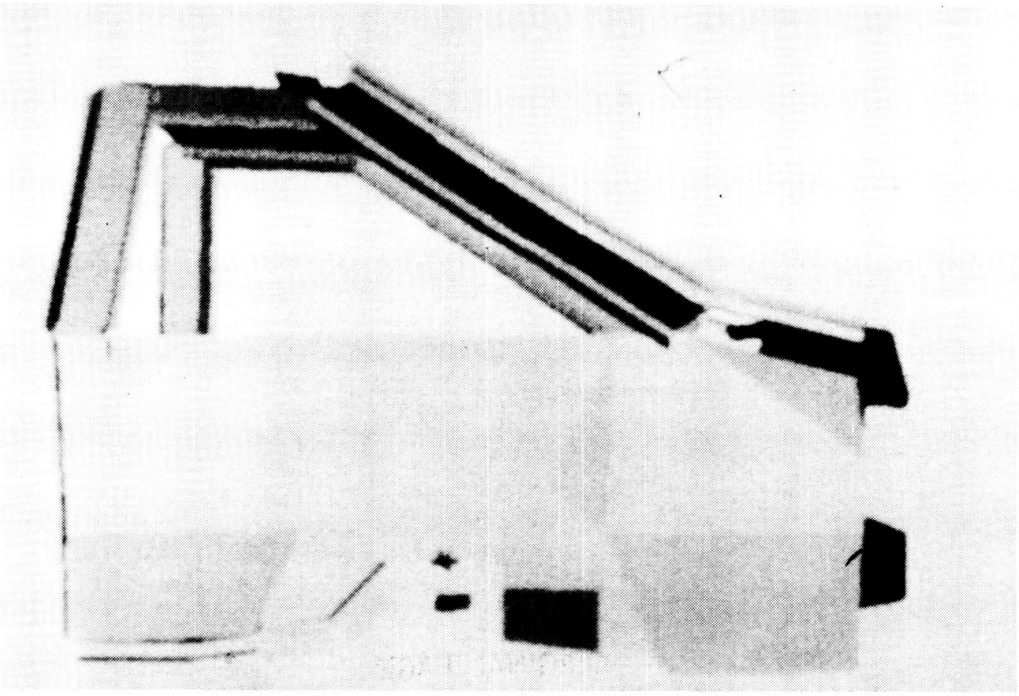


Figure 11. Undeformed model.

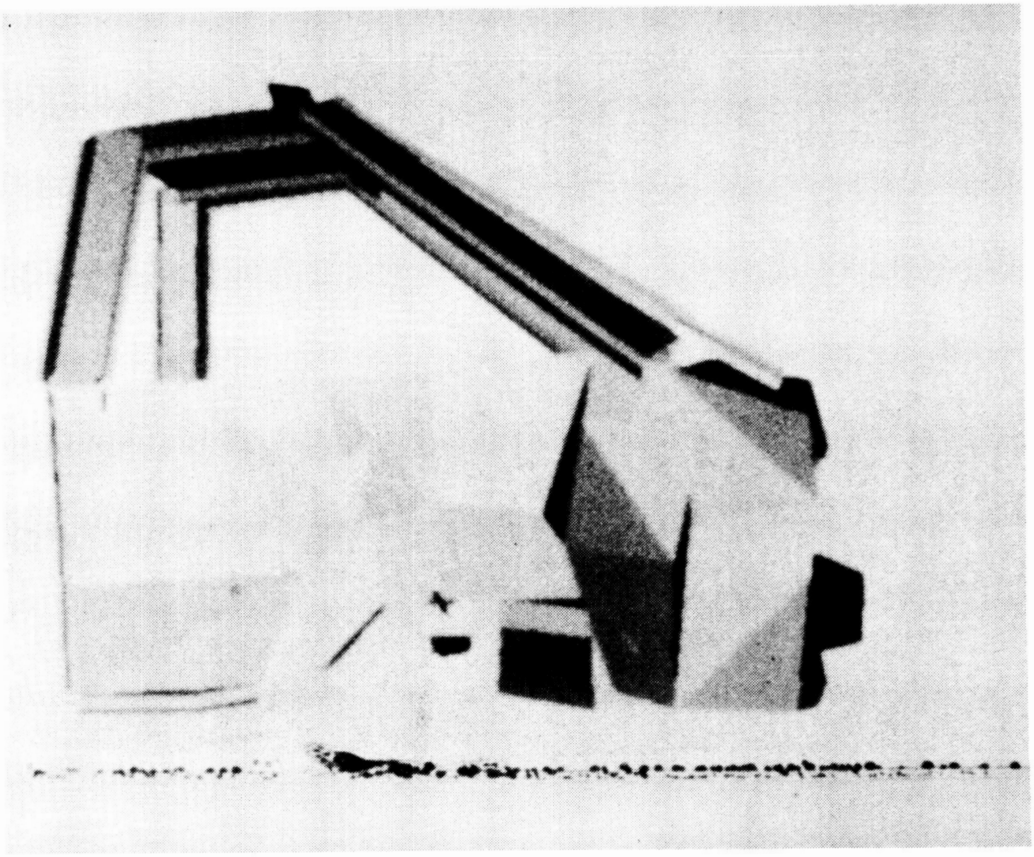


Figure 12. Undeformed model.

Experience with mathematical crash simulations has shown that, while using an adequate nonlinear dynamic computer code is essential, it is not enough. The analyst must also have some expertise in the art of modelling the structure for the nonlinear crash analysis in order to produce sufficiently accurate results within an acceptable time and cost range.

The total costs of an analysis are composed of the labor involved in creating the model and evaluating the results and the costs of using the computer. Although the modelling labor cost can be large, it is rarely discussed in the technical literature, probably because of its variability. A first-time full-vehicle finite-element model could require from one to four man-months of effort to prepare and verify, depending on factors such as the convenience of the vehicle geometry data (digital data base or drawings on paper), the use of computer graphics, and the experience of the personnel. In any case, modelling labor costs are dependent on the model size and complexity (quantity of nodes, elements, and DOF). However, after preparation and verification of the finite-element model is complete, it can be modified easily, at small cost, enabling the investigation of the effects of structural modifications.

The computational costs are dependent on model size and complexity. If it were "the best of all possible worlds" we might produce a model as shown in figure 13 for the crash analysis of an automobile. This is the type of model frequently used for linear analysis. Because of the limitation of current computers, a nonlinear dynamic analysis of this type of model is currently not feasible. However, to do so is our goal!

At the present time we consider a nonlinear vehicle crash model of 1500 DOF to be large for use on even the fastest scalar computers such as the IBM 370/3081 or CYBER 760. From two to ten restarts could be required to complete such a crash simulation. However, the new vector computers such as the CRAY-1 and the CYBER 205 allow at least a twofold to fourfold increase in overall computation speed coupled with increased memory size. In the future, improvements in both software and hardware should continue to reduce computer expense to allow more detailed models to be analyzed in smaller time periods.

Examples of computer time for two representative structures for our code are shown in figures 13 through 16.

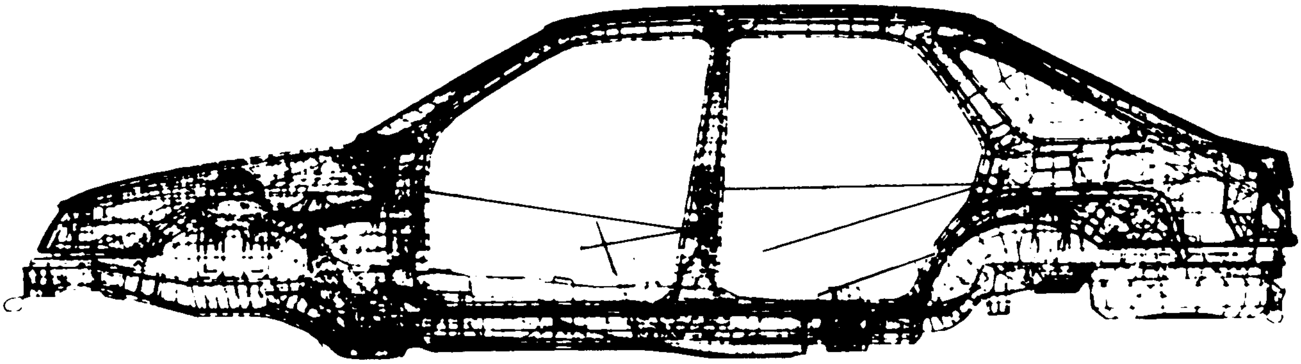


Figure 13. Detailed finite-element model of automobile.

ORIGINAL PAGE IS
OF POOR QUALITY

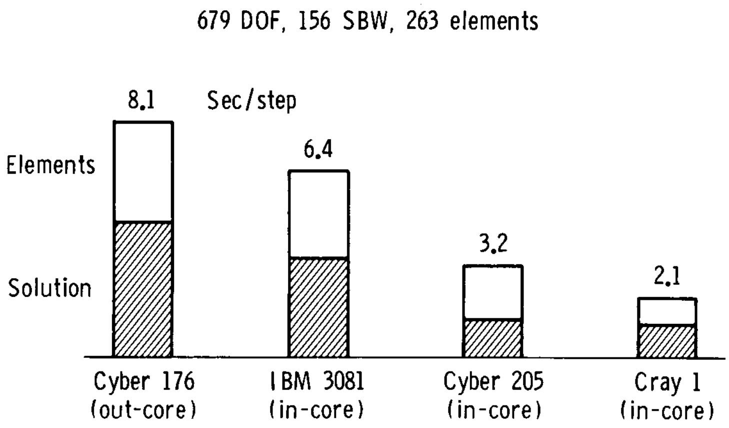


Figure 14. Scalar versus vector computers-autos rear impact.

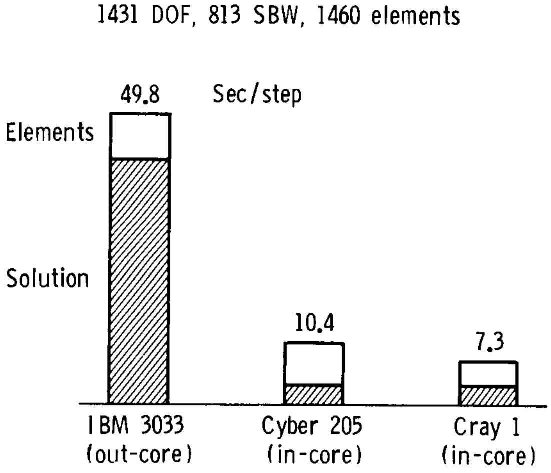


Figure 15. Scalar versus vector computer-helicopter drop.

Degrees of freedom

SBW/DOF	500	750	1000	1250	1500	1750	2000	2250	2500	2750	3000
0.10	25614	55564	97389	151089	216664	294114	383439	484639	597714	722664	859489
0.20	46889	103414	182439	283964	407989	554514	723539	915064	1129089	1365614	1624639
0.30	65664	145639	257489	401214	576814	784289	1023639	1294863	1597963	1932938	2299788
0.40	81939	182239	322539	502839	723139	983439	1283739	1624039	2004339	2424639	2884939
0.50	95715	213215	377590	588840	846965	1151964	1503840	1902589	2348215	2840714	3380090
0.60	106989	238564	422639	659214	948289	1289864	1683939	2130514	2629589	3181164	3785239
0.70	115764	258289	457689	713964	1027114	1397139	1824039	2307814	2848464	3445989	4100389
0.80	122039	272389	482739	753089	1083439	1473789	1924139	2434489	3004839	3635189	4325539

Figure 16. Core requirements for matrix assembly and solution.

In the early use of nonlinear finite-element models for crash analysis, a purely theoretical approach was attempted in which all the behavior was modelled using the finite elements. However, in the solution of practical problems involving actual vehicle structures, it quickly became apparent that some hybrid elements would be required in which the user specifies the nonlinear stiffness that is derived either from test data or a separate analysis. In the simplest case this would involve the modelling of a specific energy-absorbing component by a nonlinear spring with a user-specified crush curve. In the more complex cases, the collapse of a section of structure could be represented by a hybrid element, either because the crush test data were already available or because the nonlinear behavior of a component would be so complex and so localized that it would require too much computational effort in a small part of the vehicle.

This led to a modelling strategy in which we recognize three distinct behavioral zones in a vehicle structure when preparing a nonlinear finite-element model for crash analysis. These are linear behavior, moderately nonlinear behavior, and extremely nonlinear behavior zones. In the linear behavior zones, no nonlinear behavior is expected, and these zones are modelled as lumped masses or as rigid bodies with finite dimensions, or occasionally with a small number of deformable finite elements. In the moderately nonlinear zones, plasticity, material failure, and large deflections are expected, but the large deformations are not confined to highly localized regions. These zones are represented by a distribution of nonlinear finite elements in sufficient quantity and of the types required to allow for expected modes of deformation and failure. Here, the attempt is made to minimize the complexity while still approximating adequately the necessary stiffnesses. In the extremely nonlinear zones locally large deformations occur, such as: the collapse of a thin-wall hollow beam into accordion bellows-type folds, the complete local flattening of the cross section of a thin-wall hollow beam to form a weak "hinge" at a bend, and the collapse of a sheet metal panel into very short waves of accordion-type folds. The theoretically accurate modelling of such components requires a large number of plate elements involving thousands of DOF for each collapse zone. The added details of these local collapse models could increase the analysis costs by orders of magnitude. A practical approach for these components is to model them as simple nonlinear spring elements that require an input curve of force versus displacement or moment versus rotation. Thus, this local hybrid method requires the analyst to specify the expected nonlinear behavior. This method's great advantage is that only one DOF is added for each such nonlinear spring. However, if the conventional hybrid method is used, these nonlinear collapse curves are specified a priori without regard to the interactive effects of other loads acting in combination at the collapse zone. Since these combined loads can greatly reduce the collapse strength, they should somehow be taken into account.

In the case of a collapsing hinge forming in a thin-wall hollow beam, we have used a semiempirical interactive method involving the use of nonlinear rotary springs imbedded between beam elements in a full-vehicle model. The rotary springs are at first rigidized and the analysis using DYCAST is begun. The beam elements indicate the instant when lateral collapse begins as a plastic hinge forms. The analysis is then restarted at an earlier time with a revised moment versus rotation curve for the rotary spring element. This revised rotary spring curve rises to the collapse moment, then it decays rapidly with increasing rotation angle. The collapse moment, is determined interactively by the beam elements in the DYCAST analysis, and the shape of the rotary spring curve is taken from test experience. Typical results with this method in auto crashes predict collapse moments of hollow beams in the range of 10 to 50 percent of the theoretical fully plastic limit moment from bending acting alone.

This reduced peak moment is primarily caused by the presence of a large compressive force in the beam, acting together with the hinge moment, although the other moments also have an effect. (Fig. 17.)

- Linear zone
  - Elastic
  - Small deflection
- Modelled with:
  - Rigid bodies with lumped mass
  - Relatively few elastic finite elements
  - Substructure with most DOFs omitted
- Moderately nonlinear
  - Plastic
  - Large deflections on a global scale
- Modelled with:
  - Nonlinear finite elements
  - Allowance for possible collapse modes
- Extremely nonlinear
  - Large deflection on local scale
  - Requires fine model (>1000 DOF)
  - Special energy absorbing devices
  - Crushable nonstructural parts
- Modelled with:
  - Nonlinear spring elements
  - Spring properties from test or other analysis

Figure 17. Behavior zone characteristics.

A representative all-composite fuselage cabin section was designed, built, and crash-tested by Bell Helicopter and analyzed by Grumman using the DYCAST code. Two separate fuselages were built. One fuselage was tested in a flat drop at 30 ft/sec (9.1 m/sec) vertical velocity onto a flat, rigid surface, and the other in a 20-deg rolled altitude under the same conditions. Finite-element models of these two test cases were prepared for analysis by DYCAST, and the results were compared to those of the tests.

The fuselage section (fig. 18) was a structure composed of solid and sandwich panels made of epoxy resin reinforced by continuous fibers of graphite, Kevlar, and glass. The primary energy-absorbing structure was the honeycomb sandwich panels forming the vertical webs of the subfloor beams and bulkheads at the rear third of the fuselage under the fuel, passenger, and transmission masses. Additional amounts of such vertical sandwich material were placed in the forward subfloor forming the transverse bulkhead under the crew masses. The entire test article weighed 3530 lb (1600 kg), of which only 462 lb (210 kg) was for the structure and the remainder was from the added masses (transmission, fuel, crew, passengers, seats, ballast, cameras, and wiring).

The full cabin finite-element model is shown in figure 19. For the flat drop case, only the left half of the fuselage model was used in accordance with the symmetry of the structure and the impact conditions. The full structure model was used for the case of the impact of the 20-deg rolled attitude.

The structure was modelled with a combination of nonlinear springs, orthotropic membrane triangles, stringers, and beam elements. Nonlinear crush springs were vertically oriented within the structure to represent the crush behavior of the subfloor vertical panels of both the energy-absorbing sandwich and the nonabsorbing (breakable) type. Nonlinear gap springs controlled the impact and rebounded at the rigid ground surface.

The flat drop model contained 276 nodes, 716 elements, and 587 DOF and required 50 msec of event time, 241 time steps, and 43 CPU mins on an IBM 370/3033 computer.

Figure 20 shows a comparison of certain vertical accelerations for the DYCAST analysis and for the test. The acceleration predictions were generally in good agreement with the test data. The maximum predicted crush deformation of 4.4 in. (112 mm) in the subfloor structure was approximately 15 percent greater than that measured in the test. In addition, the deformation modes of the analysis agreed very well with those of the test.

The 20-deg roll model contained 504 nodes, 1470 elements, and 1431 DOF. It used 60 msec of event time, 760 time increments, and 450 CPU mins on an IBM 370/3033 computer. The increase by a factor of 10 in the CPU time for the rolled impact compared to the flat impact was caused partly by the doubling of the model and partly by the smaller time step required to follow some highly nonlinear local behavior.

A sampling of the data for the 20-deg roll case is shown in figure 21. The front view of the deforming structure (fig. 22) shows the crush of the lower left subfloor, the rotation of the fuselage about the impact point, and the lack of distortion in the upper bulkheads. This figure does show a distortion of the transmission mounting fixtures, caused by the inertial resistance of the transmission mass to the sideward acceleration of the roof when the vehicle was rotating. The predicted maximum vertical crush of 6.1 in. (155 mm) in the subfloor was approximately 10 percent less than that of the test. The predicted accelerations showed a mixed

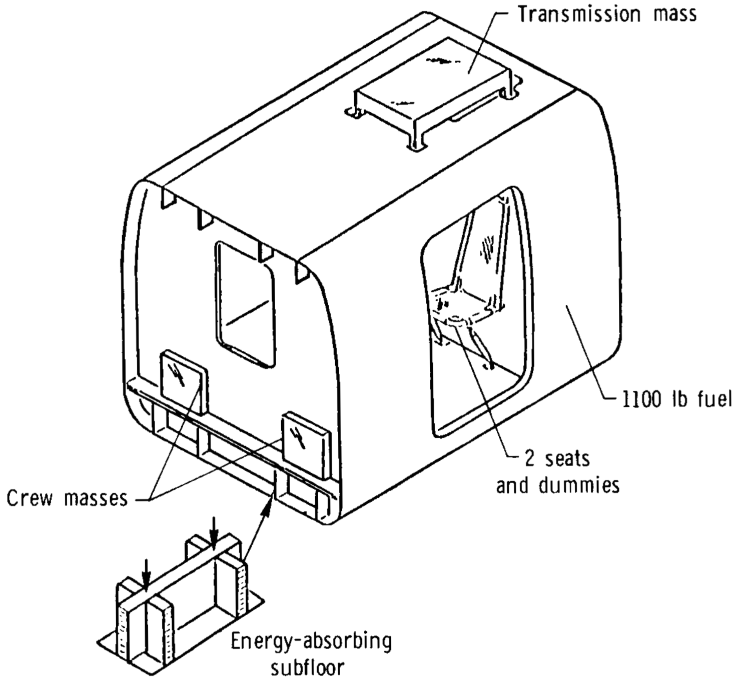


Figure 18. Composite helicopter cabin structure, external view.

FINITE-ELEMENT MODEL FOR 20º ROLL DROP

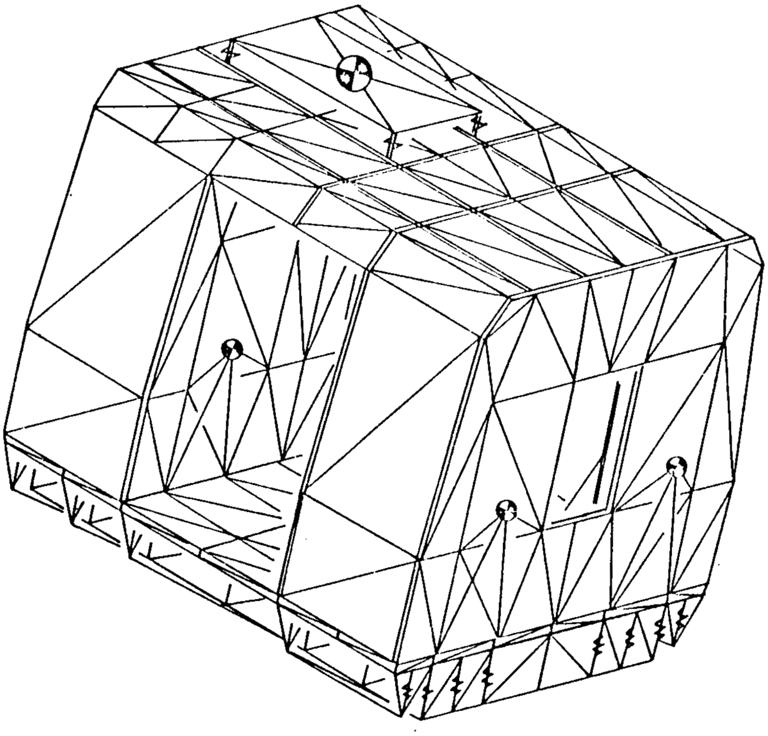


Figure 19. Finite-element model for 20° roll drop.

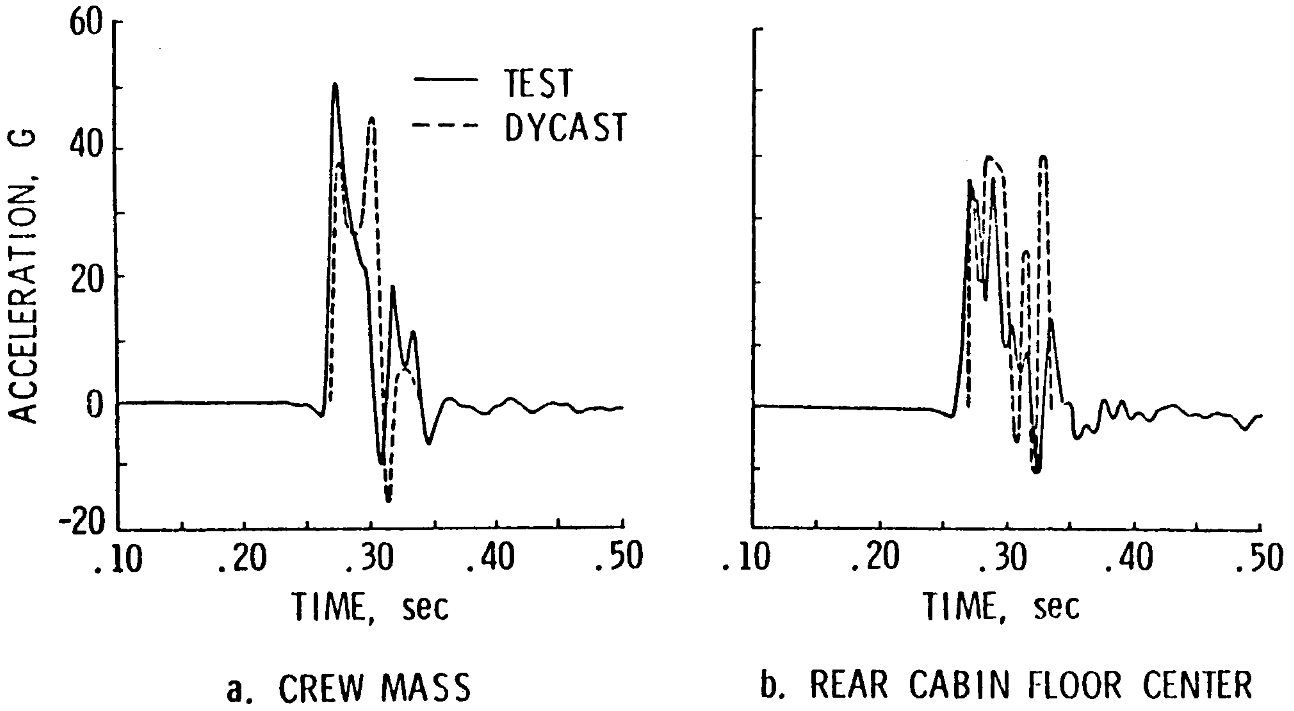


Figure 20. Vertical accelerations for flat drop.

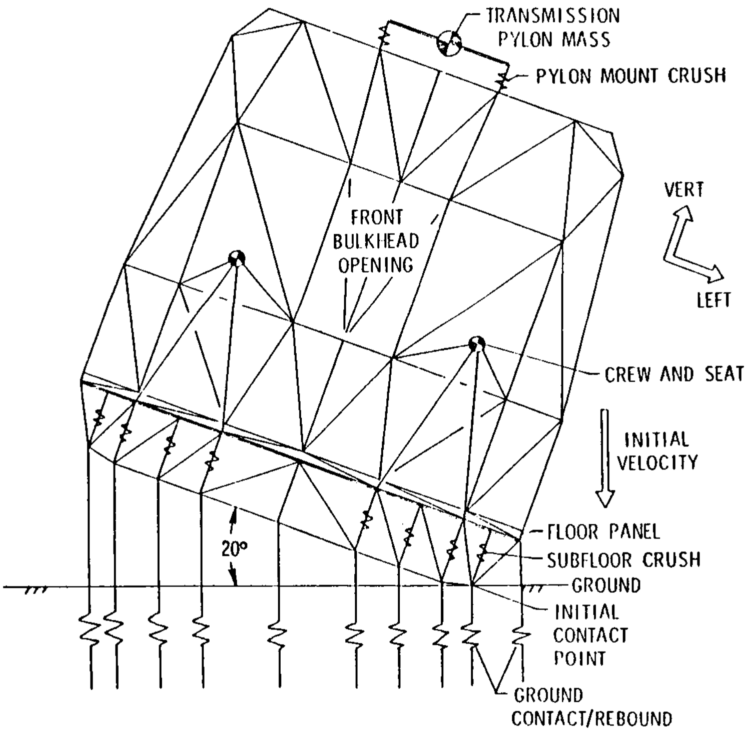


Figure 21. Front view of roll drop model.

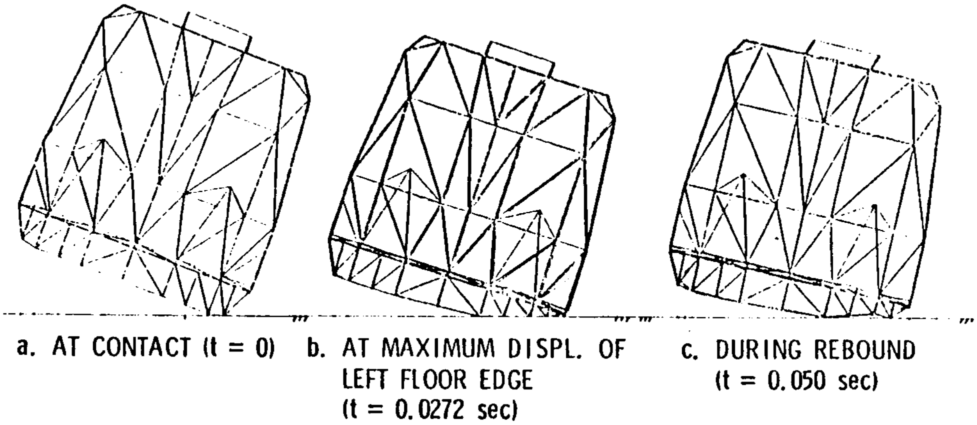

a. AT CONTACT (t = 0)
b. AT MAXIMUM DISPL. OF LEFT FLOOR EDGE (t = 0.0272 sec)
c. DURING REBOUND (t = 0.050 sec)

Figure 22. Deformed DYCAST model - roll drop.

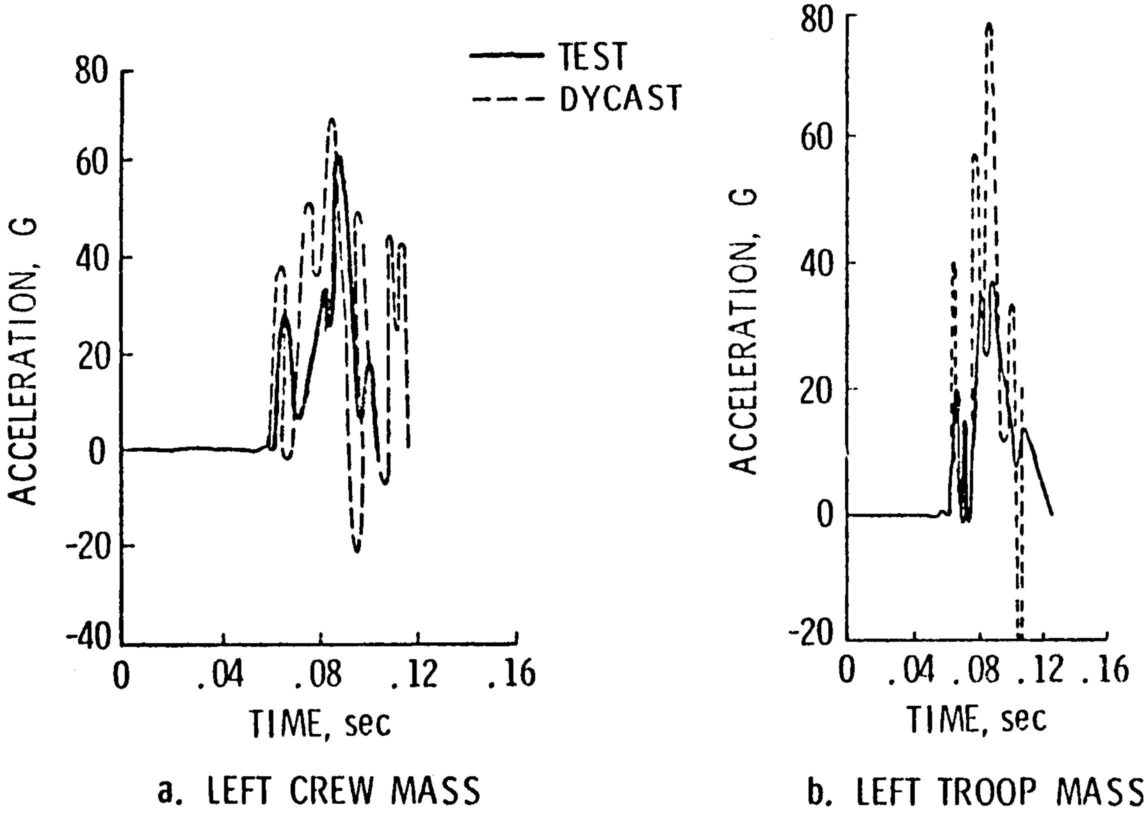


Figure 23. Vertical accelerations for roll drop.

correspondence with the test data. The left crew mass acceleration agrees well with the test data, but the left passenger mass peak acceleration is overpredicted by a factor of 2 (fig. 23).

Figures 24 through 27 outline a front barrier impact of an early prototype version of the 1984 Chevrolet Corvette, a two-seat front engine sports car with a steel frame, and a fiberglass reinforced plastic body shell. Figure 25 shows the steel frame for the analyzed vehicle, and it should be noted that the production vehicle's frame is significantly different, so that the discussion here pertains only to the early prototype and not to the final production vehicle.

The three-dimensional finite-element model involved only the left half of the car to take advantage of the symmetry. The structure was modelled all the way to the rear because it was anticipated that the engine and driveline would become a major load path to the rear of the frame (figs. 26 and 27).

The finite-element model included the frame, plus the other structure (engine bulkhead, front floor, etc.), driveline, and mechanical parts described previously. The fiberglass body was not modelled because, in the previous auto crash analysis, the fiberglass body absorbed a negligible amount (less than 5 percent) of the total kinetic energy.

The model used 157 nodes, 220 elements, and 597 DOF. The elements included 98 beams, 63 membranes, 12 stringers, and 47 nonlinear springs.

One complete simulation of 100 ms consumed 200 min of CPU time on an IBM 370/3033 computer system, required 2000 time steps using the Newmark-Beta implicit method, and was performed in four consecutive overnight segments using restart.

A complete discussion of this analysis is found in reference 1. Conclusions are found in figure 28.

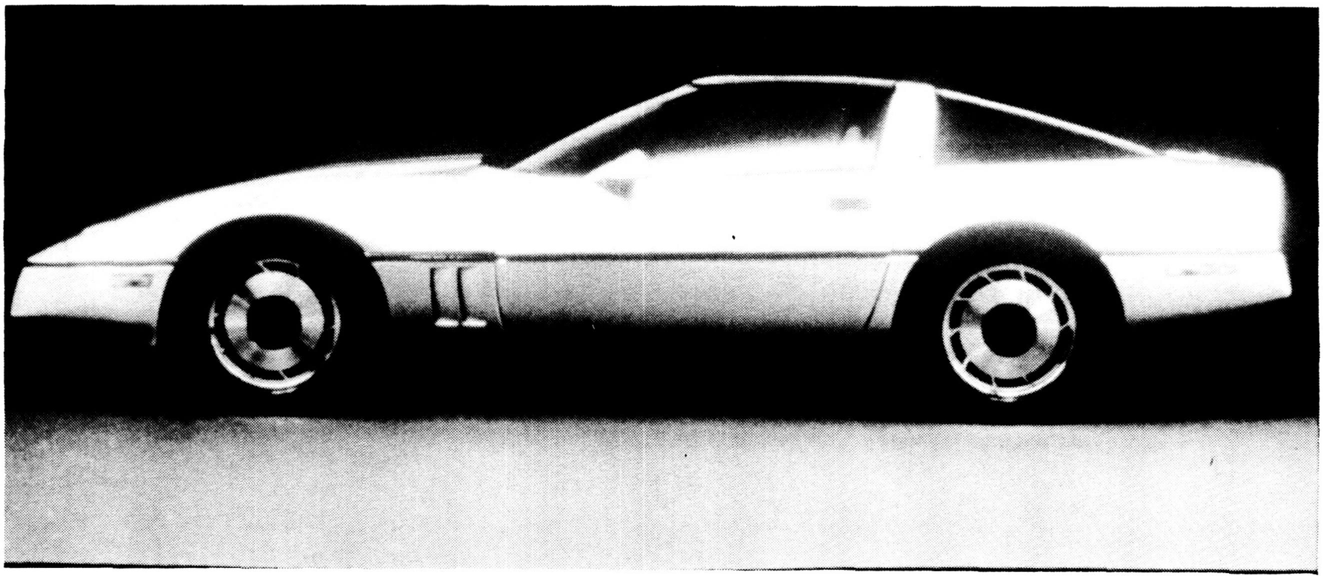


Figure 24. 1984 Chevrolet Corvette.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

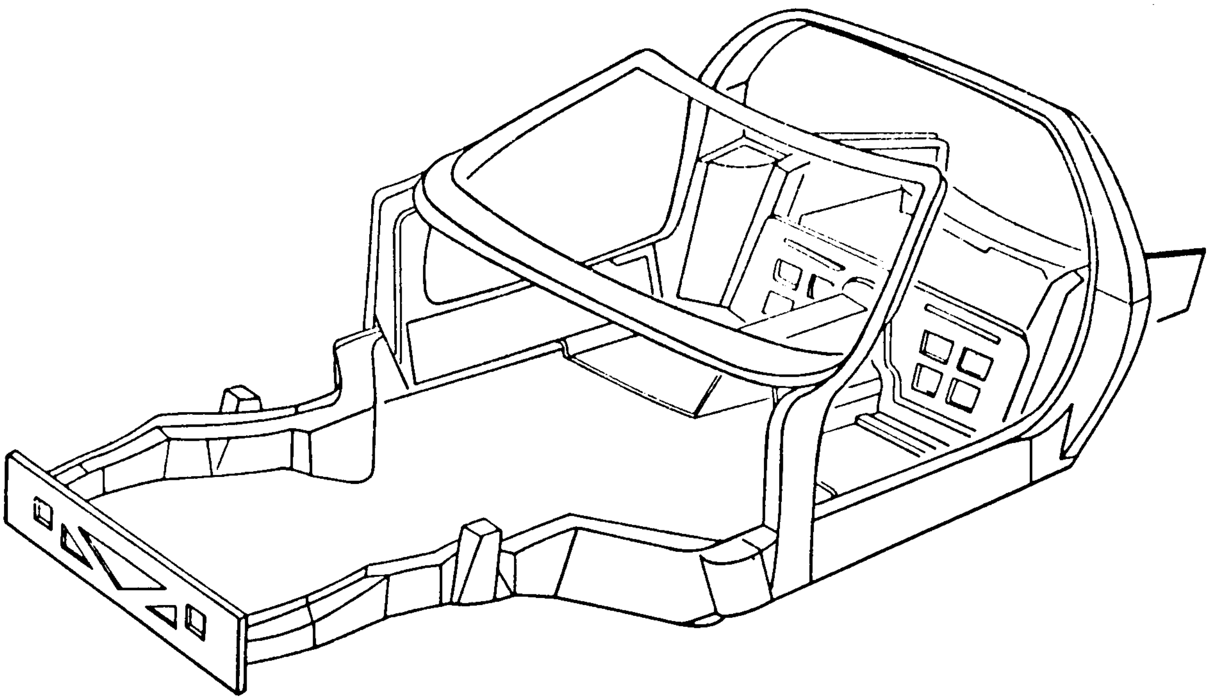


Figure 25. Welded steel frame of prototype.

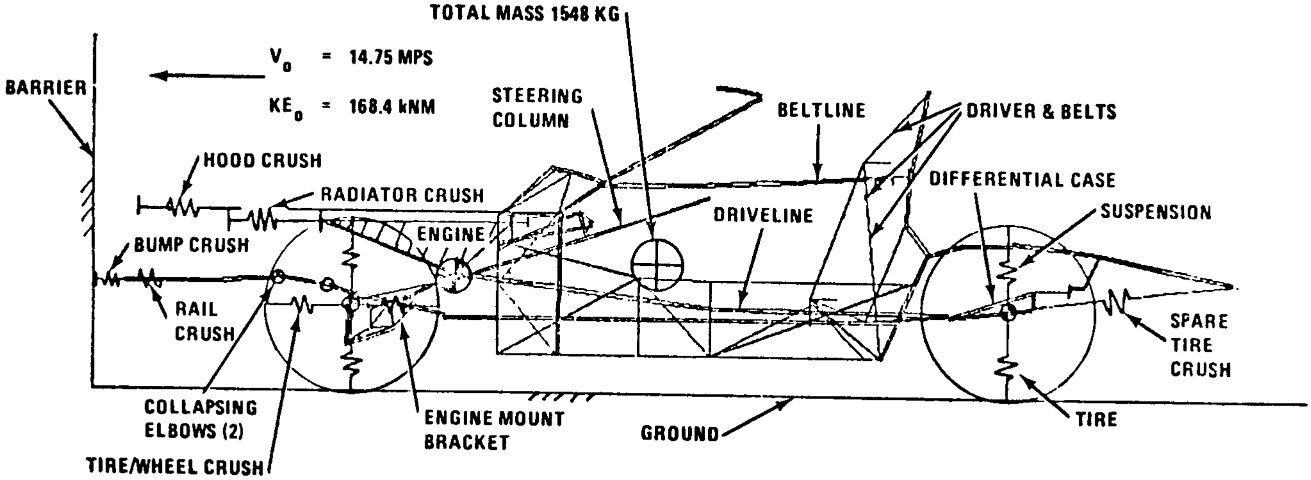


Figure 26. Side view of finite-element model.

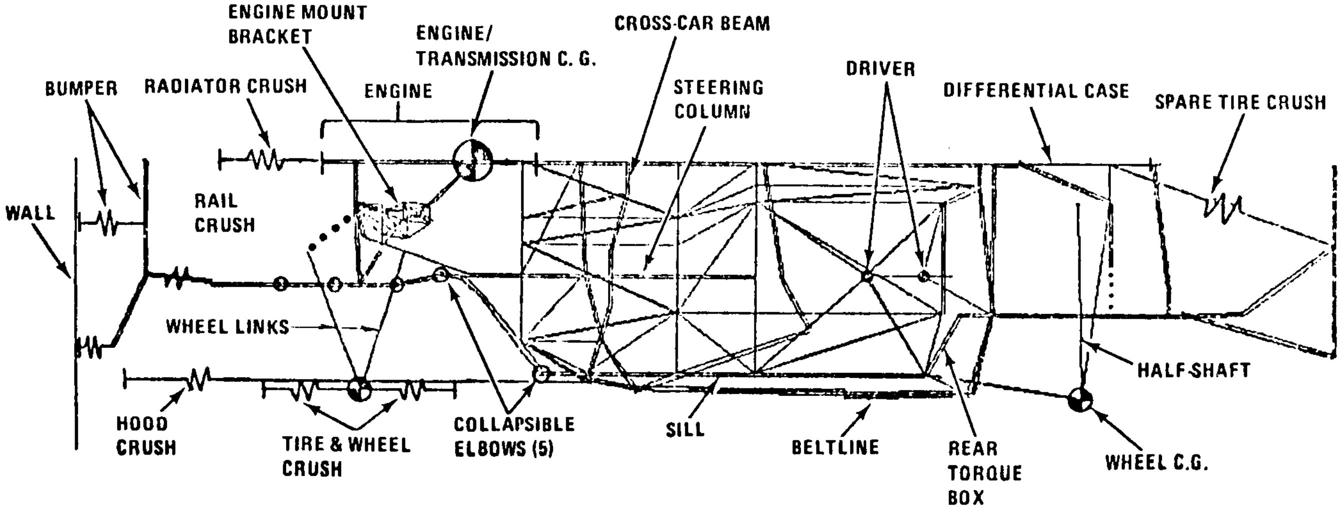
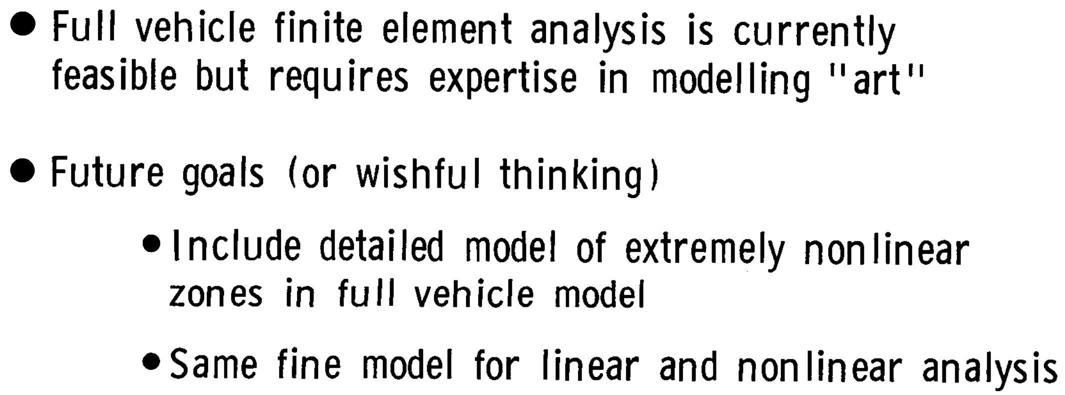


Figure 27. Top view of finite-element model.



- Full vehicle finite element analysis is currently feasible but requires expertise in modelling "art"
- Future goals (or wishful thinking)
    - Include detailed model of extremely nonlinear zones in full vehicle model
    - Same fine model for linear and nonlinear analysis

Figure 28. Conclusions.

REFERENCE

1. Winter, R., Crouzet-Pascal, J., and Pifko, A. B.: Front Crash Analysis of a Steel Frame Auto Using a Finite-Element Computer Code, SAE Fifth International Conference of a Vehicle Structural Mechanics, Detroit, April 1984.

N89-24659

APPLICATION AND IMPLEMENTATION OF TRANSIENT ALGORITHMS IN COMPUTER PROGRAMS*

David J. Benson
University of California, Lawrence Livermore National Laboratory
Livermore, California

INTRODUCTION

This presentation gives a brief introduction to the nonlinear finite element programs developed at Lawrence Livermore National Laboratory by the Methods Development Group in the Mechanical Engineering Department. The four programs are DYNA3D and DYNA2D, which are explicit hydrocodes, and NIKE3D and NIKE2D, which are implicit programs. All of these programs were originally developed by John Hallquist in association with David Benson.

This presentation concentrates on DYNA3D with asides about the other programs. During the past year several new features were added to DYNA3D, and major improvements were made in the computational efficiency of the shell and beam elements. Most of these new features and improvements will eventually make their way into the other programs. Although the latest version of DYNA3D has not been released yet, it should be available well before the end of the year.

The emphasis in our computational mechanics effort has always been, and continues to be, efficiency. Although the supercomputers of today are almost unbelievably fast, a large nonlinear finite element analysis is still superior. To get the most out of our Cray supercomputers, we have vectorized the programs as much as possible. Vectorization is not enough, however, so we must always consider the efficiency of every algorithm we implement. The net result of our efficiency criterion is we are restricted to only the simplest elements and algorithms. All of our elements have linear shape functions. We use radial return instead of a subincremental method for our plasticity calculations. Our explicit codes use only reduced integration with viscous hourglass control. Our implicit programs use quasi-Newton methods to speed convergence.

In the remainder of the presentation, several of the more interesting capabilities of DYNA3D will be described and their impact on efficiency will be discussed. Some of the recent work on NIKE3D and NIKE2D will also be presented. In the belief that a single example is worth a thousand equations, we are skipping the theory entirely and going directly to the examples.

*Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-Eng-48.

ELEMENTS

DYNA3D has three elements: an eight node hexahedron, a four node shell and a two node beam. The shell and beam elements are based on the formulations of Hughes and Liu (ref. 1). All of the elements except the beam use reduced integration with viscous hourglass control. Large strains and large deformations are assumed for all of the elements. On the Cray-1, the elements require about thirty-five microseconds per integration point for a simple constitutive model, such as the standard J_2 plasticity model with isotropic and kinematic hardening. The shell and beam models were only recently vectorized. The original implementation of the shell element required 16000 microseconds per element, which made it unusable. Vectorization alone does not account for the dramatic increase in the speed of the shell element.

Suzuki provided us with the structural data for a frame member of a car chassis along with their experimental results from a 30km/hr impact into a barrier.

Our simulation of their experiment used a mesh of 1600 shell elements. The frame member is tied at each end to a rigid body. One rigid body represents the barrier, and the other represents the sled which provided the momentum to crush the frame. The constitutive model is the usual J_2 model with isotropic, linear strain hardening. Unfortunately, the experimental stress-strain data indicate that the material does not strain harden linearly, and we believe that much of the error we see in our simulation is the result of the linear strain hardening. We are going to modify the material model to take into account the nonlinearity and rerun the analysis in the near future.

The experiment lasts thirty-five milliseconds. On the Cray-XMP/48, DYNA3D uses a little over four hours of CPU to simulate the entire event. The peak deceleration, an important number to chassis designers, which occurs at only five milliseconds, can be calculated in less than half an hour of CPU.

The results of our analysis matched the peak deceleration almost exactly, but the duration of the peak was too short. We believe that the discrepancy was caused by either the previously mentioned simplification of the material model or the 2000Hz filter that Suzuki used on their data.

Based on good accuracy of our results and their reasonable cost, we believe that finite element analysis should no longer be regarded as strictly a research tool in crashworthiness design, but as a tool for the designer.

CONTACT, IMPACT, AND FRICTION

The contact and impact algorithms have long been among the strongest points of DYNA3D. The penalty approach is used in both the two and three dimensional versions of DYNA. A distributed parameter approach is also available in DYNA2D based on the algorithms developed by others for HEMP, TENSOR, and TOODY. Aside from the obvious simplicity of the penalty approach in comparison with the distributed parameter approach, the major advantages of the penalty method are that it is symmetric and that it does not excite hourglassing modes as much as the distributed parameter approaches. The surface stiffness for the penalty method is automatically calculated based on the material properties, instead of being input by the user, which we believe accounts for the excellent reliability of the method.

We have two fundamentally different algorithms for surface contact. The original one assumes there are two different surfaces which may come into contact. They are designated the master and slave surfaces. The limitation to this approach is a surface cannot buckle and collapse onto itself. Our second method eliminates this problem, but it is slower. Most of the CPU time in the algorithms is used in the search for the regions in contact, and we have not been able to vectorize this section of code to any significant extent.

The Coulomb friction model blends the transition from the static to the dynamic coefficient of friction with an exponential decay based on the relative velocities of the contact surfaces. Several calculations performed with DYNA2D show good agreement with experimental results using this model.

EXAMPLE: Metal forming

Shearbanding was studied in this analysis. The problem is neither planar nor axisymmetric, and therefore had to be analyzed in three dimensions. A cylindrical blank of 304L stainless steel is high energy rate forged (HERF) at 1850F with shearbanding resulting in one plane. The initial velocity of the ram is 600cm/sec.

The die is modeled as a rigid body in the analysis so that a very fine mesh can be used to define the curved surfaces of the die without incurring a computational penalty from either the large number of elements or the Courant stability limit on the integration time step. The ram is also modeled as a rigid body with enough mass to give it the proper momentum. The cylindrical blank was modeled with J_2 elastoplasticity.

The analysis was run both with and without friction between the blank and the walls of the die. Shearbanding only occurred when friction was included. The contours of plastic strain correlate quite well in a qualitative way with the shearbands of the acid-etched forging.

Roughly 5 CPU hours on the Cray-1 were needed for the calculation. Higher ram velocities would require proportionally less CPU time, and lower velocities would require longer.

EXAMPLE: Pipe whip

The damage caused by one pipe hitting another is an area of interest to the nuclear power industry. In this example, a free segment of pipe collides with another section of pipe fixed rigidly at both ends. This model uses shell elements with the two surface contact algorithm. It runs in only four minutes on the Cray-1.

EXAMPLE: Axial buckling of a cylinder

This example demonstrates the use of our single surface contact algorithm for problems with buckling. The analyst does not know a priori where the cylinder will fold and therefore cannot divide the surface into a series of master and slave segments. Two contact surfaces were defined, one being the exterior surface, the other, the interior experience. The length of the cylinder is 440mm, the diameter is 100mm, and the wall thickness is 1.5mm. The mesh consists of 1980 shell elements with five integration points through the thickness. A little under thirteen CPU hours on the Cray were used in this analysis, most of which was used in the contact routine.

RIGID-BODY DYNAMICS

A recent addition to DYNA3D is material type number 20, the rigid-body material. In many crash analyses, the plastic deformation is localized to a rather small region, but the entire structure must be modeled in order to include the correct amount of momentum and inertia in the calculation. To reduce the cost of such a calculation, we replace the regions far from the impact with rigid bodies. Rigid bodies cost only one microsecond per zone, which, when compared to a constitutive evaluation, is almost free. The fact that a rigid body is defined by a material type makes this feature almost transparent to the analyst. Regions of a structure are easily frozen by switching them to material type 20. Several materials are easily merged into a single body by adding merge cards to the data file. All of the contact algorithms and most of the boundary conditions worked with rigid bodies with only minor changes to the code. In addition to the standard boundary conditions, we have implemented joint constraints. DYNA3D is probably the only hydrocode that has universal and ball joint models in it.

EXAMPLE: Earth penetrator

This calculation was run to determine the effect of a collision with a tree trunk on the trajectory of an Earth penetrator. The original calculation, which took four CPU hours on a Cray, modeled the tree trunk with an elasto-plastic material and the penetrator was elastic. We replaced the elastic material model with the rigid-body material and reduced the cost to six CPU minutes. The large reduction in cost resulted not only from the large reduction in the number of constitutive evaluations, but also from the Courant stability limit. In the original calculation, stability was determined by a small element in the projectile, but with the rigid-body projectile, stability was determined by the comparatively coarse zoning of the tree trunk. The results of the two calculations agreed almost exactly.

EXAMPLE: Cylinder impact

This test was run several years ago by Sandia National Laboratories. A steel cylinder was gripped "rigidly" at both ends by an apparatus that slammed it into a steel rail at a velocity of 1676cm/sec. The number of interest is the depth of the dent in the side of the cylinder, which is known to be 3.64cm from the experiment. This problem was one of the first successes for DYNA3D. The original calculation used an elasto-plastic model for the cylinder. An extremely dense, elastic material was used for the two rings representing the apparatus gripping the ends of the cylinder. Both the cylinder and rings were modeled with solid elements. Fifty-five CPU minutes are required for the original model, which predicts a dent 3.886cm in depth. The cost of the analysis drops to thirty-one minutes if the rings are made into rigid bodies, but the dent is only 3.048 cm. Our conclusion is that the apparatus was not nearly as stiff as everyone had assumed. We recently ran the problem modeled with shell elements, and it took only sixteen minutes; however the deformation was too large.

INTERACTIVE REZONING

Interactive rezoning has been available for several years in DYNA2D and was recently added to NIKE2D. Rezoning allows the user to eliminate or smooth sections of a mesh with thin or highly distorted elements. This increases the computational efficiency of the programs by allowing a larger time step in DYNA2D and by improving the convergence rate in NIKE2D.

This NIKE2D example shows a thick-walled cup being formed by back extrusion. The mesh was rezoned several times during the analysis. Only a few minutes of CPU was required. The same analysis with DYNA2D would take several hours because of the low speed of the forming process.

ITERATIVE SOLUTION OF EQUATIONS

One area in which we are currently supporting research is the iterative solution of linear algebraic equations. The cost of factoring a cube with N elements on each edge is proportional to N^7. An iterative solution method is faster than a direct solution for even fairly small problems of this type. For example, with eight elements on an edge, a direct solution takes 1.08 minutes of CPU while the iterative solution takes .53 minutes. With twenty-four elements on an edge, the direct solution takes 3380 seconds for the 46875 equations as compared to the 125 seconds for the iterative method.

Improvements to the element-by-element preconditioner, developed by Hughes, for the conjugate gradient method are being developed (Robert Ferencz, Lawrence Livermore National Laboratory, unpublished data). The major difficulty with iterative methods is their lack of robustness -- problems that have a wide range of eigenvalues, caused, for example, by structural elements or a penalty contact algorithm, converge slowly (if at all) with these methods. The goal of this research is the development of a preconditioner that will improve the robustness of the conjugate gradient method.

In this example, a bar hitting a rigid wall is modeled with 2700 solid elements and two planes of symmetry, giving 9196 equations. The direct method requires a little over 2.5 million words of storage, while the iterative method requires 1.7 million words. The solution cost with the direct method required 2249 seconds on the Cray XMP with 18 seconds of I/O to the solid state disk. With a standard disk, almost 1800 seconds of I/O would be required. In contrast, the element-by-element method required only 654 seconds of CPU and solved the problem without using the disk.

Although the range of eigenvalues for this problem is not as large as for a problem with beam elements, the problem is elastoplastic. This problem demonstrates that iterative solution methods are improving in robustness. As iterative solution methods improve and larger computers become available, many problems that would now be solved using explicit finite-element programs out of necessity will be solved more efficiently by implicit programs in the future.

FRACTURE AND FAILURE

Another area of research for us is fracture. Last year we implemented a tie-breaking slideline in our version of NIKE2D based on modifications to the program at South Carolina. The program was used to study the formation of chips in machining processes. We have also installed nodal constraints based on the same ideas in DYNA3D to study the petalling of sheet metal as projectiles penetrate it. In both cases, plastic strain is the fracture criterion. A smeared crack fracture model is also available in DYNA3D. An element fails when the maximum principal stresses exceed a specified fracture stress with this model. Our work in this area is very preliminary; we have concentrated more on the methods of implementing failure criteria efficiently rather than a sophisticated fracture criteria. As we gain experience and experimental results, we hope to improve the fracture criteria.

The computational overhead associated with the failure models is small. The smeared crack material model is only about twenty percent more expensive than our standard elasto-plastic material model, and the tie-breaking nodal constraints are completely vectorized.

A steel plate, .1 inches thick, is struck by a 3 inch diameter rigid sphere with a velocity of 6000in/sec. The data were chosen rather arbitrarily, with the only goal of the problem to see whether or not petalling would occur. We plan to run better calculations in the future and compare them to experimental results.

CONCLUSIONS

Given the increasing size and speed of computers and the increasing efficiency and robustness of finite-element algorithms, we believe that problems regarded by most as impossible today will be possible, if expensive, tomorrow. On the Cray-2, which is the technology of today, a multi-tasked version of DYNA3D could solve problems with more than a million zones and ten thousand time steps in less than ten hours.

REFERENCE

1. T. J. R. Hughes and W. K. Liu: Nonlinear Finite Element Analysis of Shells: Part I. Three-Dimensional Shells. J. Comp. Meth. Appl. Mech., vol. 27, 1981, pp. 331-362.

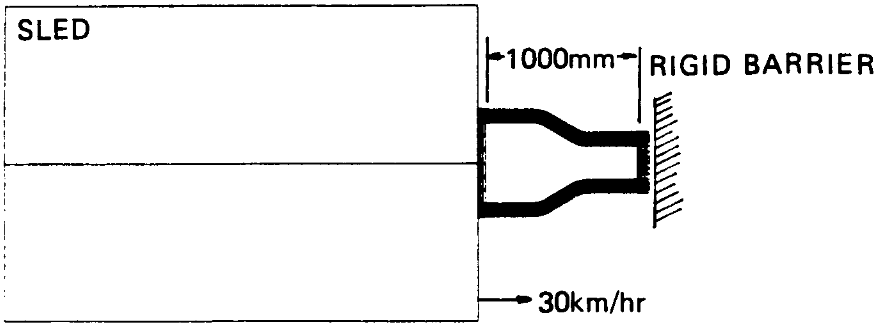


Figure 1. Finite-element model of Suzuki sled test.

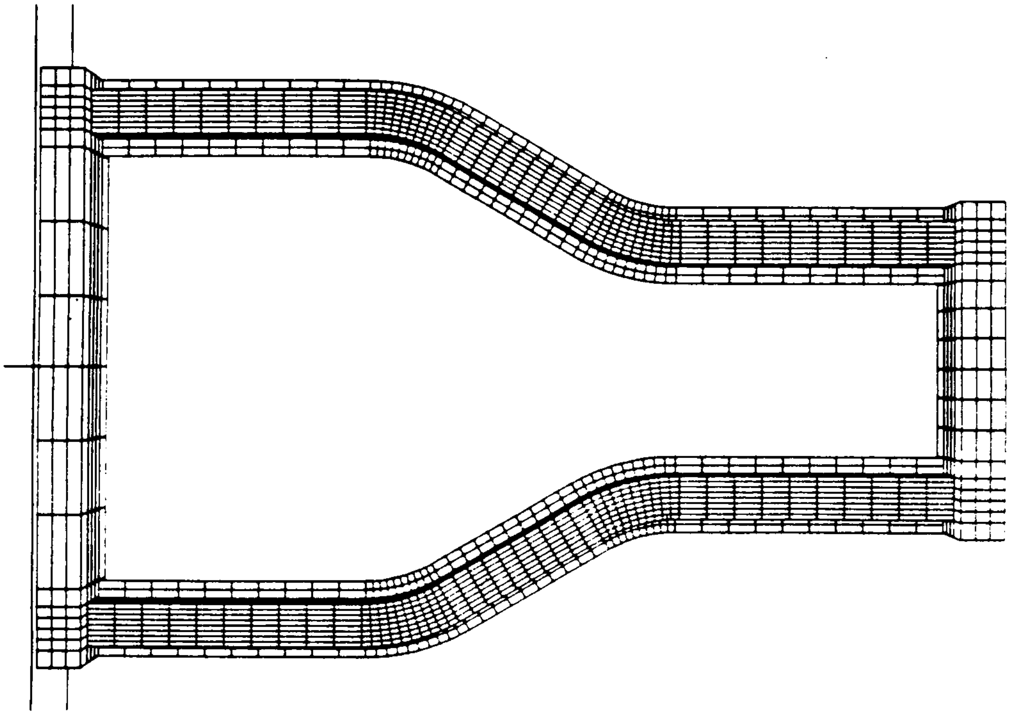


Figure 2. Close-up view of the finite-element mesh. The mesh contains 1600 shell elements.

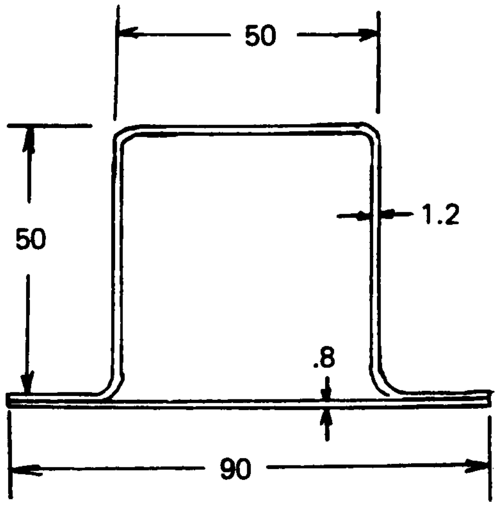


Figure 3. Cross section of frame member. All dimensions are in millimeters.

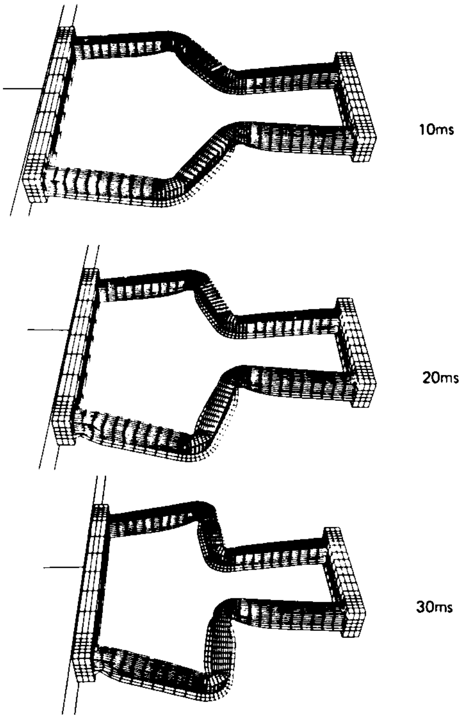
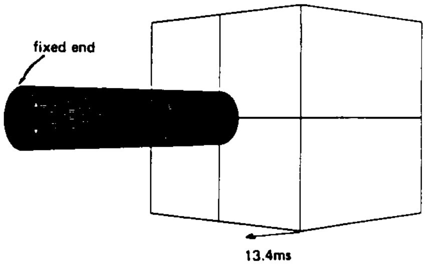
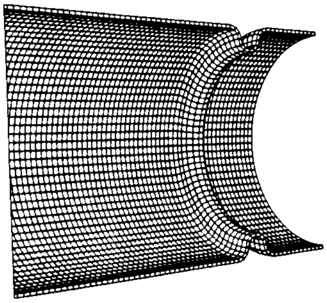


Figure 4. Deformed shapes at 10 ms output intervals.



(a) Initial configuration of cylinder.



(b) Close-up view showing cross section: One quarter of the cylinder and mass were modeled with symmetric boundary conditions.

Figure 5. Axial buckling of cylinder.

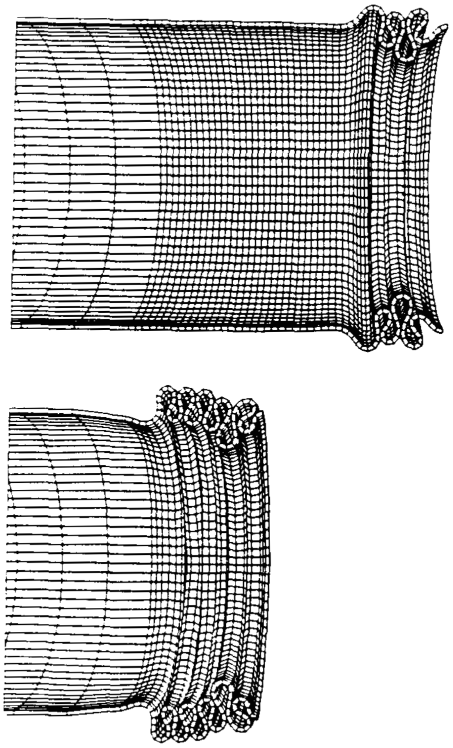


Figure 6. Deformed cross sections at 5 and 10 ms.

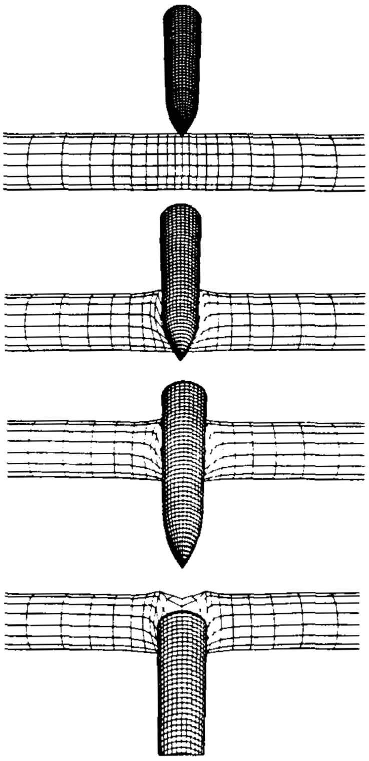


Figure 7. Earth penetrator.

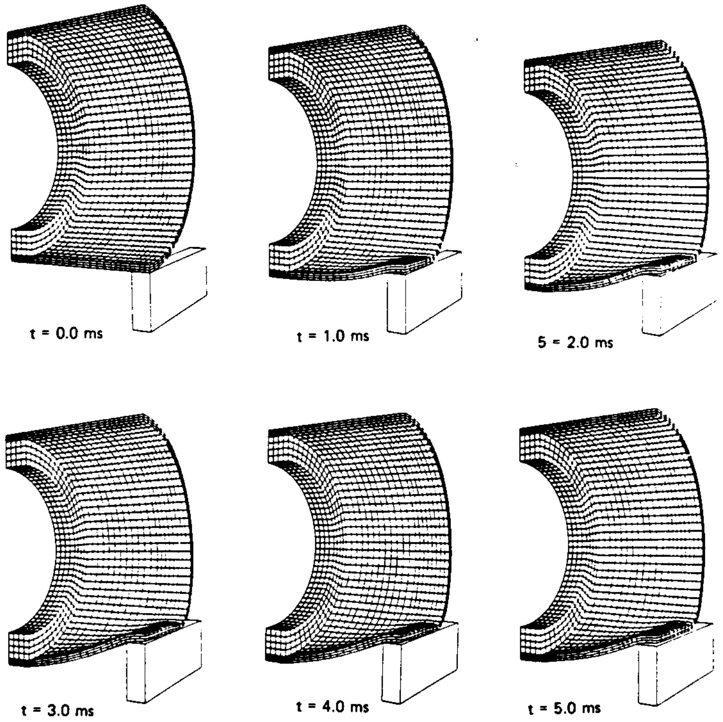


Figure 8. Cylinder impact.

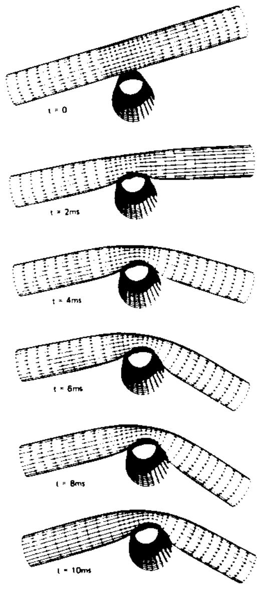


Figure 9. Sequence of deformed shapes in pipe wip analysis.

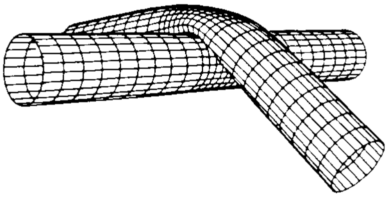


Figure 10. Rotated view of final configuration.

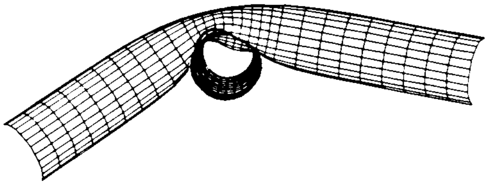


Figure 11. Deformed cross section.

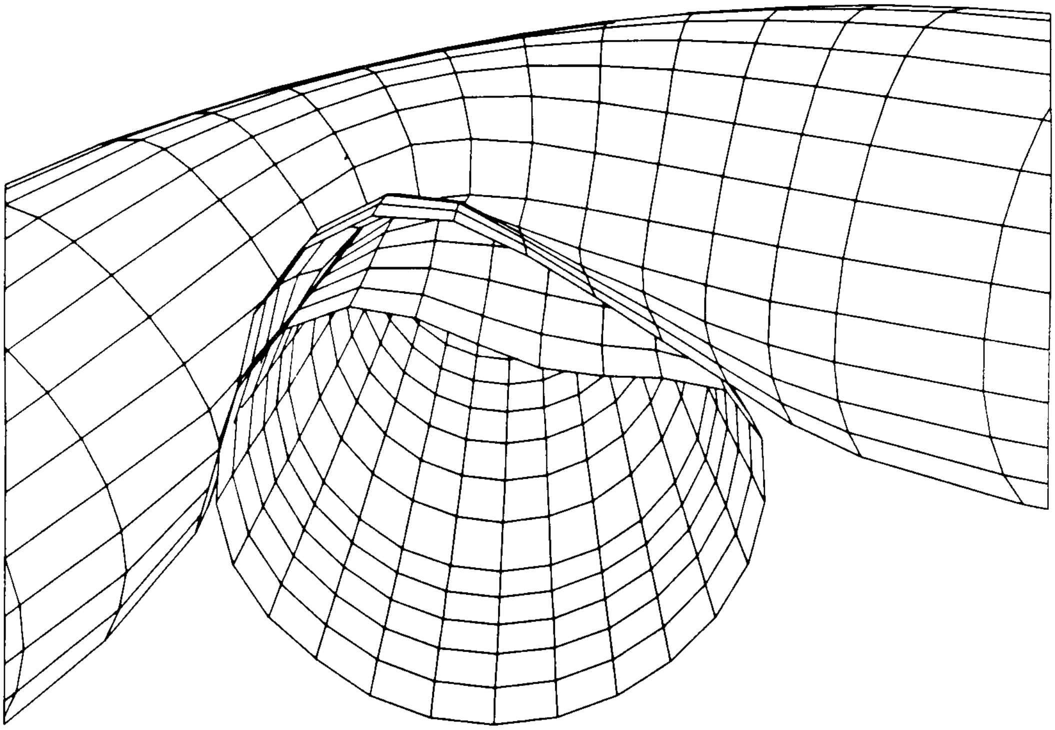


Figure 12. Close-up of deformed cross section.

QUESTION AND ANSWER SESSION
TRANSIENT DYNAMICS

Questions and Answers following: "Application of Transient Analysis to Aircraft Structures," by R. J. Melosh

Ted Belytschko, Northwestern University: I'd like to amplify one point you made and perhaps take issue with one of your statements. You pointed out that we can simulate an automobile crash and an aircraft crash and, as a matter of fact, you cited some papers in 1971 and 1973 as evidence of that.

R. J. Melosh: 1983.

Belytschko: 1983, OK. But, nevertheless, I think that if you look at the basic phenomena that are involved, they're extremely complex. I think we're being a little bit optimistic if we believe that we can analyze that class of phenomena effectively. We may be able to replicate certain salient features, but if you look at problems like dynamic post-buckling, which is an important ingredient in the aircraft problem, I don't think we can consider that problem solved because we cannot really make predictive solutions with reasonable accuracy. Furthermore, if we consider features such as joints and other aspects of the problem, I think we're just at the fringes of learning how to deal with them.

Melosh: Well, there may be a problem with respect to modeling, I'm not assessing that. But, of course, if we're going to assess the problems of modeling, then we do need the tools to evaluate the impact of idealization errors. I think we have the mechanics technology in the software. We may not know how to use it properly, but error controls and error measures could give us the information we need and the insight to do that.

Questions and Answers following: "Explicit, Implicit, and Hybrid Methods" by Ted Belytschko

Tom Moyer, George Washington University: I was wondering why more use hasn't been made of higher order algorithms both in the implicit and the explicit class. We've done some calculations, and there are a lot in literature that suggest that you can save significantly on whatever piece you are doing explicitly or implicitly if you go to a higher order technique. It's a lot more work, but it saves you a lot in the long run. And I don't see much coming out in the finite element literature.

Ted Belytschko, Northwestern University: In explicit methods, if you compare any higher order method to the central difference rule, the benefits are going to be quite small, if not negligible because spatial discretization error dominates temporal discretization error when you are at the Courant stability limit. So in explicit methods, there's very little to be gained by trying to achieve better temporal accuracy.

In implicit methods, that is not the case. And as a matter of fact, I think that the reluctance to pursue higher order implicit methods probably stems from the fact that higher order methods require more core storage, which is already a problem with implicit methods, particularly in nonlinear problems where one has to store the hessian matrix and all the state variables. To do this for three or four historical steps, as required in higher order methods, becomes a substantial burden. If somehow one could use slow and fast memory in an optimal way, one could take advantage of the higher accuracy of these methods. I think there may be some potential for higher order implicit time integrators, because we are finding accuracy problems with conventional integrators. People have overlooked that for a long time, but as Bob Melosh indicated, many solutions that are being generated with large time steps are very inaccurate and there are no guidelines as to whether they are accurate or inaccurate.

Questions and Answers following: "Unconditionally Stable Concurrent Alqorithms in Transient Finite Element Analysis" by Dr. Michael Ortiz.

Jerry Housner: I notice you kind of pulled the structure--the subdomains--back together at the end of the analysis. Is that done at the end of the entire analysis or periodically throughout the analysis?

Ortiz: It's done at the end of every time step. In other words, during one application of the algorithm, you can process all the subdomains concurrently and information is only exchanged at the end of the time step, which limits the communication between the processors to a minimum. So that's another nice feature of the algorithms. And this information has to be done essentially for the process of averaging--the extended predicter. So that's only done at the end of every time step.

Joop Nagtegaal, MARC Analysis Research Corp.: That's very interesting, Michael. What you seem to be doing, if I just think of it for a moment on an element-by-element or subdomain-by-subdomain basis, you independently integrate subdomains, and you pull things together, right, by essentially applying conservation of momentum. Is that not what it is? Because you divide by the average mass of the points?

Ortiz: It's not exactly momentum averaging rule, it's mass averaging rule. And that's only a particular choice of many possible choices of concurrent algorithms. Any choice of an effective stiffness matrix that satisfies consistency and unconditional stability will do perfectly well. So, the method is very general.

Nagtegaal: In that process of pulling stuff together, however, I have the feeling, that you are, indeed, destroying energy somehow. Is that correct--that you put artificial damping in your system at that point and that's what makes it tick, makes it unconditionally stable for large steps?

Ortiz: Yes, well one thing's for sure, conservation of energy is not guaranteed. However, it doesn't blow up in any way, either...

Nagtegaal: If it's not guaranteed, it must somehow be disappearing, right? It must not be able to generate it, for sure.

Ortiz: There is some damping in the algorithm, yes, although a very slight one as the numerical results that I showed tend to indicate.

Nagtegaal: One more question. Ted gave a nice talk and told us how poor implicit algorithms perform, certainly for problems where you want to consider high frequency where you get tremendous phase errors, which is known. How does this method compare to the implicit method in that respect? Are they really even worse?

Ortiz: Well, as I said, the implicit algorithm is a particular case, if you choose only one subdomain. If you choose two subdomains, you're going to get something which is very similar to an implicit scheme. So accuracy would only deteriorate very slightly. As long as the number of subdomains is reasonably small for a large mesh, the algorithms are going to be very similar in accuracy to implicit schemes. Now if you go to element-by-element partitions, then it's basic to combine the algorithm with some kind of automatic time stepping technique to make sure that you are within reasonable bounds of accuracy. Then the method becomes reliable and, as I showed, can be advantageous with respect to the explicit scheme. But it is really basically these semi-implicit schemes that address the issue of accuracy which is the critical one. Unconditional stability is rather easy to obtain as you saw; the critical thing is whether the algorithms have reasonably good accuracy.

Moktar Salama, Jet Propulsion Laboratory: I have two questions. The first one is the execution time estimates that you give here for 2-D and 3-D. For example, in the dynamic case, you show that the execution time is $\frac{1}{n^{3/2}}$, n being the number of subdomains, and also the number of processors. Is that correct? Or the number of processors is n^2 or what?

Ortiz: Typically, in an implementation on a multiprocessor machine, one would probably go to a number of subdomains equal to the number of processors. Now in those estimates, there is no reference made to the efficiency of the communication network which is, in itself, a function of the number of processors. Those are ideal estimates for a 100% efficient communication network.

Salama: OK, and so there are n processors, presumably?

Ortiz: Typically, yes.

Salama: In this case I'm confused because how could you get more than an n speed-up from n processors? It doesn't make sense.

Ortiz: As you divide the mesh into n subdomains, you reduce both the number of degrees-of-freedom and the bandwidth. If you identify the cost of the algorithm with the equation solving effort that you do at the local level, you have to compute not only the reduction in degrees-of-freedom there but also the bandwidth. Now, if you multiply those two together, that's how you get the squares there, you see.

Salama: Thank you.

Editor's Note: In his calculations which produce a speedup greater than n, Dr. Ortiz is defining speedup as the ratio of ideal theoretical run time for a problem with one subdomain running on one processor to the ideal theoretical run time for that problem divided into n subdomains running on n processors. Part of the speedup is produced by dividing the problem into n subdomains; part is produced by exploiting parallel processing--running on n processors. The part of the speedup associated with parallel processing is n.

Questions and Answers following: "Transient Analysis Techniques in Performing Impact and Crash Dynamic Studies" by Alan B. Pifko

W. J. Stroud, NASA Langley: Alan, how do you arrive at the spring constant for the nonlinear springs?

Pifko: We've done it in a couple of ways. One way is through tests of the individual components--like cruciform sections of the helicopter floor. In the case of automotive rails that collapse into an accordian fold, engineers in the automotive industry bend up these models, ad infinitum. Automotive engineers develop those spring constants through many tests that they perform, both statically and dynamically. The trick then is, you're assuming a deformation mode, and you have to make sure that this assumption is correct. You have to make certain that all the energy is absorbed in the mode assumed.

Question and Answers "Application and Implementation of Transient Algorithms in Computer Programs" by David Benson

J. Tinsley Oden: Tell us something about your friction law that has three coefficients of friction and depends on velocity.

Benson: We have a static coefficient of friction, dynamic coefficient of friction, and then we have an exponential decay between the two that's dependent upon the magnitude of the relative velocity. And so you have three coefficients that you can work with. The exponentional decay factor, the static coefficient of friction, and the dynamic coefficient of friction.

Oden: Do you ever have any numerical problems with that? It's basically dynamically unstable.

Benson: No, we haven't.

Oden: Did you ever do any simple problems like sliding a block down a plane, or something like that?

Benson: Yes, that was the toboggan case and it worked fine.

Oden: All right. You must have some artificial viscosity in there somewhere because that's a very unstable algorithm. You're feeding energy into the system when you do that. Whenever the coefficient of friction decreases with an increase in velocity you're generating energy in the system. Perhaps we can talk about it later.

Benson: OK.

TRANSIENT DYNAMICS PANEL DISCUSSION

COMMENT, D. J. WEIDMAN, NASA LANGLEY RESEARCH CENTER: Perhaps we can get some good responses from the attendees and speakers concerning the dynamics problems that have been presented and the approaches that have been discussed. To help do this I prepared one viewgraph.

COMMENT, ROBERT MELOSH, DUKE UNIVERSITY: Entry number three in your viewgraph asks if computers make intelligent choices. I'd like to dispel the idea that a computer can do anything intelligent. The computer can have access to information that the analyst doesn't have and, in that sense, it can make surrogate choices or choices that the analyst might not make.

COMMENT, D. J. WEIDMAN: The question in item three was only intended to address whether the computer could pick between different algorithms (i.e., whether they're explicit, implicit, or a hybrid).

COMMENT, ROBERT MELOSH: Computers, of course, are all equally accurate. The question is really one of efficiency, and I'm not sure the decision is that simple. If we could teach someone (a graduate student, for example) to make an intelligent choice, then we could teach the computer to make an intelligent choice.

COMMENT, TED BELYTSCHKO, NORTHWESTERN UNIVERSITY: Because parallel machines are now becoming commercially available, we have a tremendous opportunity in time integration. In contrast to linear statics, for example, one finds more substantial payoff in computing time when utilizing parallel architectures. We really haven't exploited that aspect of computers, and I think it's very important that this be done early in the game in regard to the development of the hardware. I've heard it said that running time is not that important; if we need an answer badly enough, we'll wait for it. I think, however, that if we are going to utilize transient analysis in a computer-aided environment, we need quick feedback. Actually, what happens is that when an engineer wants an answer to a question and it means waiting overnight, the question is often discarded. Consequently, obtaining quick running time is essential in introducing transient nonlinear analysis into the (computer-aided engineering) CAE-type environment.

COMMENT, DAVE BENSON, LAWRENCE LIVERMORE LABORATORY: I have one remark concerning what are considered supercomputers today and what they are capable of. Cray 1 is about 10 years old and no longer deserves the title of supercomputer; the new champion is the Cray 2, which will perform over a billion floating point operations per second and has 67×10^6 words of memory. With that kind of capability you can now take codes and run million-element problems with 10,000 explicit time steps and get it done overnight. That accomplishment is feasible with today's codes on today's machines.

QUESTION, D. J. WEIDMAN: Do we have any remarks, questions, or observations from the audience?

QUESTION, EDWARD HAUG, UNIVERSITY OF IOWA: I have a question for either Alan Pifko or Dave Benson. I'm scared half to death with the size of these problems. I had convinced myself that we mechanical engineers in mechanisms and machines were within 15 years of the state of the art, but I'm not so sure now. It's embarrassing to talk about, but there is an article in Business Week* on how computer-aided engineering is going to revolutionize the design process. This article suggests that engineers are going to have a CAE system on their desks by 1988 and that this system will do a multitude of things. As a matter of fact, the picture that leads into the article is Alan Pifko's collapse of the forward vehicle. I guess I have a hard time imagining that by 1988 every engineer is going to have a Cray 3 or 4 on his desk. You're going to have to have specialists: Would you comment on how these massive, high-speed codes can be used?

COMMENT, ALAN PIFKO, GRUMMAN AEROSPACE COMPANY: What we're doing is certainly not, at this point, computer-aided engineering. However, we can use the existing model (for example, one that contains finite and hybrid crush elements) as a resource. If engineers can at least use a Cray 1, which is 10-year old technology, then they might be able to ask the question: What happens if we make a configuration change to the model? For example, if we have a two-door car, what happens if we change to a four-door car? What happens if we put a hatchback on the car? If we already have a model and a Cray 1, we can do a sensitivity study. I don't think that it is feasible to start from scratch, to compose a model in real time, and to run it immediately so that it works. I do think, however, that the sensitivity study is feasible on a CYBER 176. We saw that the crash model ran a couple of hours and really impacted the computer system. If we run on a Cray 1 we can put it in the computer and come back after lunch, and it will be completed. This accomplishment has changed the way we do business.

COMMENT, DAVE BENSON: I don't think there are going to be Crays on engineers' desk tops by 1988. I also don't think that the average analyst is going to be producing huge models either. Many analysts today don't even use finite-element codes, as common as these codes are. Finite elements are essentially entities unto themselves. It takes an immense amount of data to really produce a challenging problem for a supercomputer. There are meshes that we use for DYNA 3-D in which the data file itself is 120,000 lines long. Obviously those meshes are computer generated. But the average analyst is not going to have the ability to generate those either. And there won't be a need for every analyst to crash a car model into a wall. There are too many analysts involved with other things such as designing the dash accessories and so on. So there's really not that much demand for it. There are important and bigger problems to be solved in the future, but for the average analyst, I don't think there's going to be that dramatic an impact.

COMMENT, TED BELYTSCHKO: I think it's easy to take a negative viewpoint on what's going to happen as far as nonlinear analysis. For example, if you were to consider finite elements 20 years ago, then doing linear analyses of complicated structures by finite elements looked like a very formidable task which, perhaps, would always be removed from the mainstream of engineering. If you go to a plant like General Motors today, you find that linear finite-element analysis is a very standard task that requires almost no background in finite elements. It is done in a matter of minutes. Essentially they have a data bank with a car. One can call back any part of that, put a light pen around it, produce a finite-element model, and do vibration analysis or linear static stress analysis. This is something that is routinely done by numerous engineers in General Motors. And, I think, although perhaps not evident for

*"Tests by Computer Make Trial-and-Error Old Hat," Business Week, June 17, 1985, pp. 144H-144J.

modeling a car crash, there is a tremendous interest in companies like General Motors to do nonlinear analysis associated with manufacturing processes, for example, in designing a tool die or forging process. It is true that today there are great impediments. Number one, we still do not have enough ability to transfer the data bases and number two, there is a lot of expertise involved in running a nonlinear program. If we can overcome those obstacles, however, I think nonlinear analysis will play a much larger role in engineering in a very commonplace setting. In other words, a person who designs a die will not be an expert in nonlinear finite-element analysis, but he will use a nonlinear finite-element analysis to see what kind of strains are being developed in a sheet during a sheet metal forming process. This may not be evident in 3 years, but in 10 years, the impact will be substantial in nonlinear analysis.

COMMENT, ROBERT MELOSH: I agree with Ted. There is no point in restricting our imaginative view of how far analysis can go. I think that would be the wrong view to take. I think we should project larger analyses for the future, and even if we don't get there from an applications point of view, the analyses that are run today will be better.

COMMENT, JOHN HEDGEPETH, SANTA BARBARA, CA: I would certainly subscribe to that last viewpoint. I think that all of us will make a mistake if we try to minimize what might happen in the future. I don't think we can predict exactly what's going to happen, but the future is going to bring things that are a lot different from what they are today. I would like to remark on item three in the viewgraph (can computers make intelligent choices?). When I see these models which involve lots of large deflections, strains, and motions, and when I see contact problems, they only involve a very small percentage of the entire structure. Many of the models had thousands of grid points, most of which were really unexercised in the actual problem that was carried out. It would be nice if the computer could make the intelligent choices of putting the grid points closely spaced in those areas in which the physical behavior was going to require close grid spacing. Now, if we have intelligent graduate students, or even intelligent engineers, those engineers have learned somewhere along the way that when they have a problem that exhibits a boundary layer, they'll put grid points more closely spaced in such areas where the boundary layer is and then economize by making the grid spacing larger in other areas. That's an intelligent engineer. The problem that the engineer has in modeling these nonlinear problems, such as wrinkling of membranes, is where the regions are that require these close grid spacings. It would be nice if the computer could sense what is going to happen and at the time that some local nonlinearity occurs, automatically refine the local mesh. I would consider that to be of great service to us in treating these problems.

COMMENT, ROBERT MELOSH: I guess I have to defend my position. I don't exclude an adaptive mesh refinement; I think that's very important. I think that adaptive mesh refinement is an essential thing. I don't consider that an intelligent activity though. I could assign an undergraduate student to that kind of activity.

COMMENT, JOHN HEDGEPETH: I don't think you can, and let me tell you why. The difficulty is you have a coarse mesh, but the loads have gotten to the point where crippling is imminent. Neither the finite-element code that you're running nor your undergraduate student knows that the structure wants to cripple unless you give it more intelligence to know that. Now I do agree that you've got to tell the computer the same thing that you've got to tell the undergraduate student so that a decision can be made. If for instance you're studying the buckling of a tube, depending on the wall thickness, that tube plastically deforms into nice axisymmetric folds or into diamond pattern folds. It would be nice for the computer to tell the engineer whether the deformation will be axisymmetric folds or diamond pattern folds. There is something occurring in the material that is making the tube deform one way or the other. It would be nice if we could know how to tell all of that and have the computer do that work for us.

COMMENT, D. J. WEIDMAN: That's a good point John. Are there some other questions or discussion points? Another discussion item is a means by which we want to address benchmark problems. This has come up before, for example at AIAA, but if you've got a new approach and you want to try it on some other programs and write it up for other people to use, how do you compare that new approach with what's already existing? It may be faster on one machine than on another; it may be faster in your code than it is in my code. And that leads us to benchmark problems. Do we have to define some as we were discussing at the recent SDM conference? AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics, and Materials Conference (SDM), Sheraton-Twin Towers, Orlando, Florida, April 15-17, 1985; Session 16: Finite-Element Standards Forum). How should we approach that? How can we compare things intelligently?

COMMENT, MOKTAR SALAMA, JPL: I do not think we need benchmark problems simply because future machines are going to be more efficient for different kinds of problems in a different way. An approach that may work well for a vector machine like Cray or a CYBER may depend on certain characteristics of the algorithm that another machine, like a true parallel machine or an array processor, might not require. It might require different characteristics of the algorithm, so I don't think it is even desirable to have benchmarks. Those machines are going to have special traits, and in order to use the machine most efficiently, you will have to exploit those traits. You design approaches for specific machines. In this sense, you have no way of defining a set of benchmark problems.

COMMENT, TED BELYTSCHKO: I would partially like to second your remarks in that the word "benchmark" and words that were used at the SDM conference (such as "finite-element standards") struck a lot of fear in people who thought that such an approach would lead to standardization and thus hinder creativity or new developments. On the other hand, if one considers performance, there is a need to benchmark various finite elements. It is very difficult for a conventional user to identify problems that will adequately test the performance of certain classes of finite elements or algorithms. I see a lot of papers that are written which propound new elements, and it turns out that if the developer had chosen an intelligent set of performance problems, he would have been able to identify right away that it would fail certain very important classes of problems. The reason for this is, of course, that unless a lot of effort is made to develop a spectrum of problems which should be within the purview of a given element or a given algorithm, it's very difficult for one person in the emergency situation (which usually arises when checking a new code) to really devise this. If these performance problems are available, I think it will serve a good purpose in the finite-element and the analytical community.

QUESTION, JOE PADOVAN, UNIVERSITY OF AKRON: That sounds good, but when you go to commercial software, how many new elements have really diffused into it from the academic arena or from the various grants?

COMMENT, TED BELYTSCHKO: Quite a few.

COMMENT, JOE PADOVAN: Quite a few? I would venture to guess that the amount is really very small.

C-2

COMMENT, TED BELYTSCHKO: I differ with that quite strongly. Look at most of the current codes, for example, the Hughes-Tezduyar* element is in the MARC code and is widely used there. NASTRAN has had a complete transition of elements over the last 20 years. As a matter of fact, I was talking to Bob Harder, and it turns out that they do not even include the documentation for C^1 elements in their standard users' manual anymore. The element is in the code but it is not documented; it is only available for old data sets. So one finds in many of the codes a tremendous transition in the types of elements which are used. There are a few codes which are still relying on very old elements. But even, for example, in Abacus, which relies very much on the four-node C^0-type elements, DKT (Discrete Kirchoff Theory) is being installed.

COMMENT, JOE PADOVAN: I agree; I don't mean that point. I'm saying that for the numbers of different types of elements that have been proposed in the thousands of papers published in the literature, the majority of these are not getting into commercial codes. Somehow there is an unconscious benchmarking process occurring. It's very difficult on a developmental level to really benchmark properly.

QUESTION, TED BELYTSCHKO: What do you mean by benchmarking a problem?

ANSWER, JOE PADOVAN: Most of the elements you described - did they really go through a very strong benchmarking processing before they were implemented? Did they go through a series of five or ten diagnostic problems?

COMMENT, TED BELYTSCHKO: I think that is standard in most software houses today. The major shortcoming in many software houses has been that they have not had a catalog of problems which would weed out all deficient elements. It's standard to run through 100 or 200 problems even for minor changes in an element in a major software house today. The difficulty has been that even among the 100 or 200 check problems, there isn't one that will catch certain deficiencies. For example, membrane locking is quite prevalent in a lot of curved elements and was not caught because there are very few problems with pure bending response in these check problems.

COMMENT, JOE PADOVAN: We'll discuss that later.

COMMENT, K. C. PARK, LOCKHEED PALO ALTO RESEARCH LABORATORY: Since this has been a dynamics session, I'd like to bring up the issues concerning dynamics. I think two of the outstanding problems, even in basic transient algorithms, are accuracy and step size control for implicit algorithms. If you know the problem and what you are looking for from it quite well, you can get very decent solutions relatively quickly. On the other hand, if you don't know the predominant physical phenomena, the nonlinear dynamics problem is not a trivial thing for an engineer to crank out. That's why many people, particularly in industry, fall back on explicit algorithms even though that's sometimes quite expensive. They know it's much easier to control accuracy and step size in explicit algorithms, so they're willing to pay the price because they know they can count on the results. I think there is a need for benchmark problems

*T. J. R. Hughes, and T. E. Tezduyar: Finite Elements Based Upon Mindlin Plate Theory With Particular Reference to the Four-Node Bilinear Isoparametric Element. J. App. Mech., vol. 48, no. 3, Sept. 1981, pp. 587-596.

in which someone can evaluate proposed new algorithms and new strategies in the transient dynamics area. This would also permit people to determine in what application a proposed application functions best.

COMMENT, JOHN HEDGEPETH: I think of benchmarks in two ways, as being an accuracy measure and a time measure. However, I get the idea from listening to this discussion that accuracy is not really the problem, it's time. When I think about the computer time it takes to do this, that, or the other thing, I get a mental picture of a very interesting number. It is the ratio of the number of dollars that are spent on computer time, divided by the number of dollars that are spent on the hours of the engineer's time, involving just those engineers who are working with the computer in solving a problem. I would venture to say that right now that ratio is probably less than 0.1. I don't know how long it will take until we have a ratio greater than 1 so that the problems that the engineers are solving are so time consuming on a computer that it really becomes an economic problem about whether the thing runs this fast or that fast. I think the most important concern is the question of how long does it take the engineer to get the thing running, and when he gets the results back, how long does he study those results before he's able to answer the problem that he is really trying to address.

COMMENT, BAHRAM NOUR-OMID, LOCKHEED, PALO ALTO: I think with the emergence of concurrent processors, we'll see the ratio of 0.1 that you mentioned go even further down because the time taken to program some of these computers is going to get even larger. I think that what we are trying to compare here is more like drivers and cars. It's very hard to say whether one person is a better driver than another; similarly it is just as hard to say whether one program is better than another when you haven't specified the car and the computer they are operating on. I think that if we can determine which of two people is the better driver, then we can determine which is the better program.

QUESTION, TED BELYTSCHKO: Could we have a clarification? Did you say the ratio of manpower relative to computers was 0.1? (Addressed to John Hedgepeth.)

ANSWER, JOHN HEDGEPETH: I said computer dollars were divided by manpower dollars.

QUESTION, TED BELYTSCHKO: Did you interpret it the same way? (Addressed to Nour-Omid.)

ANSWER, BAHRAM NOUR-OMID: Yes, I thought he meant computer dollars over man dollars. I think the cost of computers is going down, and the time taken to utilize those computers is going to go up.

QUESTION, TED BELYTSCHKO: You were talking about software development cost?

ANSWER, BAHRAM NOUR-OMID: Yes.

COMMENT, TED BELYTSCHKO: That's usually considered part of the computer cost.

COMMENT, BAHRAM NOUR-OMID: If the engineer's time is included as an overall time, then the usage time doesn't change, whereas the software developer's time is going to triple or quadruple; I think that increases the ratio.

COMMENT, ALAN PIFKO: Let's look at linear analyses subject to what you mentioned. At Grumman, the type of linear analysis necessary for aircraft type structures is a major thrust. In order to use computer-aided design and computer-aided engineering you go right from drawings to a finite-element analysis. You send it off for analysis. You sit and wait. You see the status. It comes back. And you immediately do postprocessing graphics. That's a major thrust that has gone on for linear analysis. That thrust is now moving into nonlinear problems as more and more detail, such as fracture analysis problems and contact problems, are being performed. In many major companies that is a major thrust. Ted Belytschko mentioned that it is going on at General Motors, and I've seen it also going on at Ford and Chrysler. And the nonlinear problems will now fall within that realm too as they become more deterministic and easier to use.

COMMENT, BJORN F. BACKMAN, BOEING MILITARY AIRPLANE COMPANY: I want to address this question of economy again. To us, the question of computing enters in two ways. One way is the economy and one is the safety aspect. When you look at the economy part, I don't think you can avoid including software development costs as part of the cost for computing. When it comes to the safety part with the emergence of new materials and the new requirements like damage tolerance and so on, the natural tendency has to be towards more computing. There is no way around it. You take the natural lack of forgiveness (what you call stiff composites here) and you'll find that the sensitivity to local effects simply raises the need for analyses in the design process an order of magnitude. The same thing is true for the data volumes. We may be used to the thought that, for a modern jet airliner, less than 10 percent of the cost is engineering and, of that, maybe less than 5 percent is the finite element. You're going to see the safety drivers as indirect costs making up the difference.

COMMENT, ROBERT MELOSH: I'd like to make an observation on the future. In particular, it appears to me that what we're going to see happen is that analyses are going to get more difficult for the analyst unless something is done. I think that it's fair to say that graduates of our current educational program in the universities are competent to handle linear analysis. I don't think it's fair to say they are competent to handle nonlinear analysis. I think the subtleties of nonlinear analysis are very pervasive, and the interpretation of results is very complex. Maybe I'm overwhelmed by nonlinear analysis, but I think the problems are difficult, extremely difficult, and to throw them at a computer and come back and interpret the results is an extremely hard challenge to me. I worked for MARC Analysis for 4 years, and I answered the phone for about 2 years of that time in response to problems that people had in interpreting results. Often, there were problems with reading the manual or problems with using the program. Sometimes there were bugs in the program. But too often the problem was that people just did not have the background to understand the phenomena. They had no experience with nonlinear phenomena--pogo sticking, large-scale divergence problems-- problems that just were out of their camp. And I think it's incumbent upon the community, if we expect people to use computers more and use them for problems that they're not prepared for, to hopefully provide some basis for preparation.

For example, I had a student who came back to the university from working for a year, and I asked him what he had been doing. He replied that he had been doing dynamic analysis (piping analysis) for nuclear reactors. I commented that I didn't remember that he ever had a course in flexible body dynamics and asked him how he was fairing. "Well," he said, "there's somebody else in the company that has run the code before, and they tell me how to prepare input." "But that analysis result is your responsibility. How can you fulfill your responsibility?" I asked. He replied, "I'm doing the best I can." I think there are a lot of people out there who are going to be in that situation if we provide these tools without the kind of responsibility that John is suggesting, the responsibility to protect the analyst from gross mistakes.

MULTI-BODY DYNAMICS

PRECEDING PAGE BLANK NOT FILMED

RECENT DEVELOPMENTS IN DEPLOYMENT ANALYSIS SIMULATION USING A MULTI-BODY COMPUTER CODE

Jerrold M. Housner
NASA Langley Research Center
Hampton, Virginia

Why Deployment Dynamics Analysis?

Deployment is a candidate mode for construction of structural space system components. By its very nature, deployment is a dynamic event, often involving large angle unfolding of flexible beam members. Validation of proposed designs and conceptual deployment mechanisms is enhanced through analysis. Analysis may be used to determine member loads thus helping to establish deployment rates and deployment control requirements for a given concept. Furthermore, member flexibility, joint free-play, manufacturing tolerances and imperfections can affect the reliability of deployment. Analyses which include these effects can aid in reducing risks associated with a particular concept. Ground tests which can play a similar role to that of analysis are difficult and expensive to perform. Suspension systems just for vibration ground tests of large space structures in a 1 g environment present many challenges. Suspension of a structure which spatially expands is even more challenging. Analysis validation through experimental confirmation on relatively small simple models would permit analytical extrapolation to larger more complex space structures.

- Deployment: A Candidate For Space Station Construction
- Deployment Is a Dynamic Event
- Design And Concept Validation
  - Determination of Member Loads
    Deployment Rate
    Deployment Control
  - Reliability of Deployment Mechanism
    Flexible Members
    Joint Free-Play
    Tolerances and Imperfections
  - Ground Tests Difficult and Expensive
    Suspension System in 1 g Environment
    Size Limitation

Candidate Multi-Body Programs For Deployment

Shown in this chart is a list of some of the existing U.S. computer programs which are candidates for performing deployment analyses. These programs perform multi-body dynamic analysis. Some of these programs were orignally designed for mechanisms, while others were designed for satellites with appendages. Most of these programs are in a constant state of improvement and most have or will soon have capability for treating flexible members and perhaps sophisticated joint models. However, efficient simulation of a deploying structure with a large number of components will require considerable further development.

ADAMS --------- Mechanical Dynamics

ALLFLEX ------ Lockheed Missiles and Space

CAPPS -------- TRW

DADS ---------- University of Iowa

DISCOS/NBOD -- Martin Marietta

IMP ----------- University of Wisconsin

LATDYN ------- NASA (pilot code)

SNAP ---------- General Dynamics

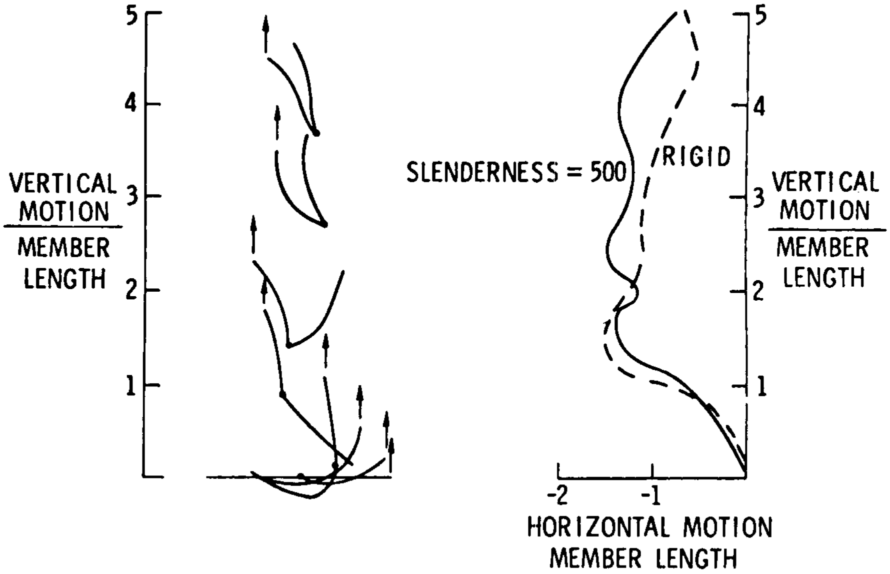
TREETOPS ------ Honeywell & DYNACS
& CONTOPS

Cambridge Research Associates Code

OTHER

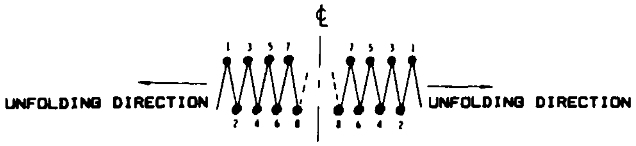
Large Distortion And Motion Of Two Pin-Connected Beams Subject To A Vertical Tip Step Load

This chart displays the time-lapse response of a generic large motion/large distortion maneuver. Two very flexible beams which are pin-connected at their common end are acted upon by a vertical step load at the free end of one pf the members. Note in the left-hand figure that the pin-connected end first moves downward before moving upward. Also note the large relative angular motions of the members and their distortions. The right hand figure shows the trajectory of the point of load application for both flexible and rigid member cases.

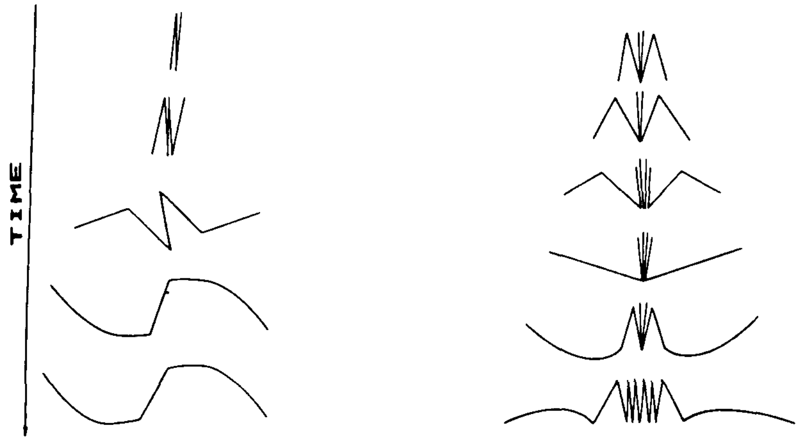


Unfolding Of Multiple Hinged Flexible Beams

This chart displays the unfolding of an accordion-type assemblage of flexible members. The members are hinged together and the deployment is driven by pre-wound torsional springs at each hinge. The deployment sequence of both a five member collection and a sixteen member collection is depicted. Due to the odd number of members in the left-hand portion of the chart, the collection of beams appears rotated. This appearance is explained by an appeal to the conservation of angular momentum. In the right-hand figure, the members are seen to deploy in a near sequential pattern. This is the natural way this system opens up and is not due to a preset adjustment of the driving springs. Rather, the closer a member is to the center of the system, the greater the mass it must push in order to open up. Hence the outer members deploy first and a near-sequential deployment pattern results.



GENERAL UNFOLDING PATTERN

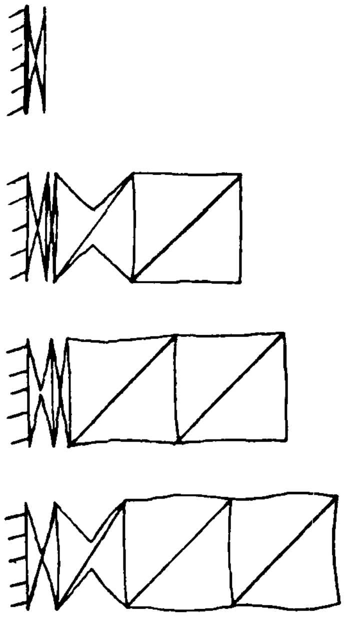


FIVE MEMBERS

SIXTEEN MEMBERS

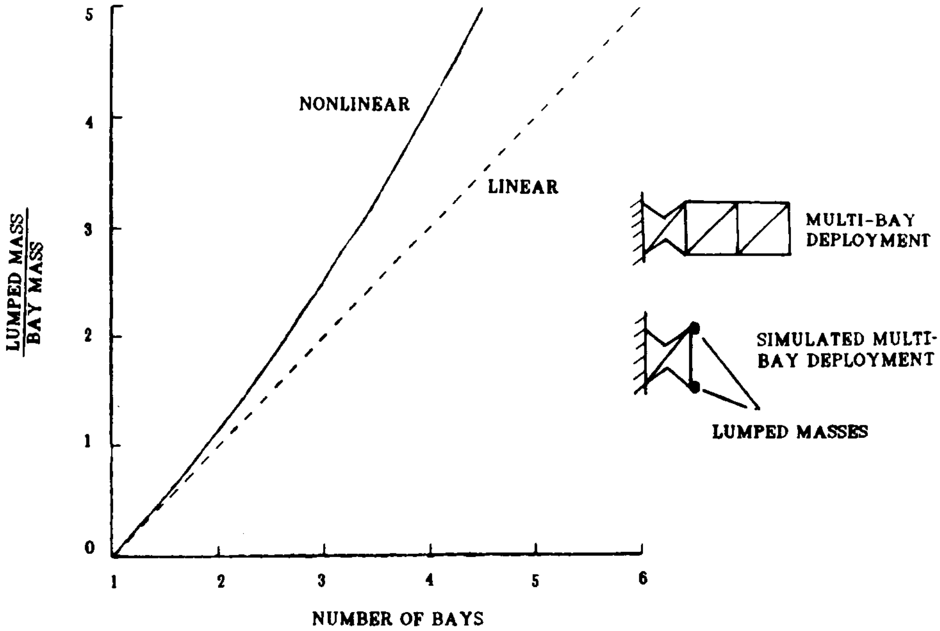
Uncontrolled Deployment Sequence of Four-Bay Mast

In this chart, the analytically simulated deployment of an uncontrolled four-bay mast composed of flexible members is shown. (The analysis was performed using the NASA LATDYN computer program and involved 64 degrees of freedom.) The deployment occurs due to unfolding of the longerons of each bay which have lockable joints midway along their length. The diagonals are assumed to telescope out during the deployment and the deployment is driven by precompressed rotational springs at each lockable joint. Typically such masts are controlled to deploy sequentially, that is, one bay at a time, but an uncontrolled deployment sheds light on the natural deployment character of the design. Moreover, insight is gained into the simultaneous deployment which can occur in other deployables such as a tetrahedral truss. The chart shows that the mast tends to deploy nearly sequentially without control. This appears to be due to the larger inertial mass which must be pushed by the inner bays and to the choice of the spring constants driving the deployment. Thus sequential deployment for a mast tends to be a natural process.



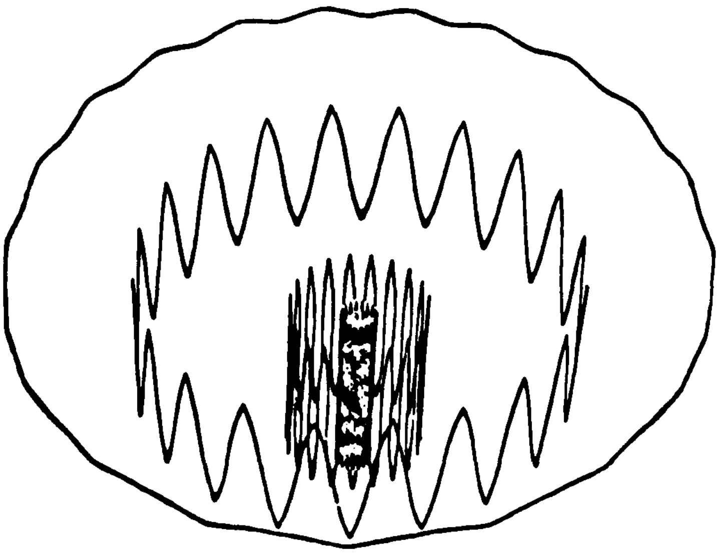
Lumped Mass Necessary to Simulate Uncontrolled Multi-Bay Deployment

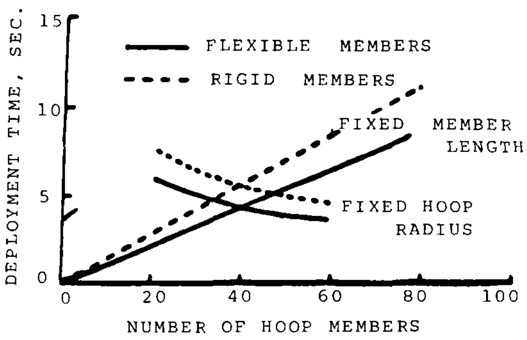
Due to the large computational time requirements of the mast deployment in the previous chart, it becomes desirable to simulate the multi-bay deployment using only one bay with lumped masses representing the inertial effect of the remaining bays. The figure shows the amount of lumped mass needed to simulate the deployment time of the multi-bay analysis. The linear curve represents the use of a lumped mass equal to the number of simulated bays. The nonlinear curve indicates the predicted mass needed for this simulation. The linear representation becomes increasingly inaccurate as the number of bays simulated increases and the added mass for multi-bay simulation must be increased.



Uncontrolled Deployment of Flexible Member Hoop

Deployment of a hoop composed of 40 flexible hinged members is considered in this chart. The left-hand figure depicts the hoop deployment sequence. Bending of the hoop members is observable. The right-hand portion of the chart indicates the variation of hoop deployment time with number of hoop members. Two sets of curves are shown. In one set of curves, the length of the hoop members is fixed so that as the number of members increases, the hoop radius also increases. In the second case, the hoop radius is fixed so that as the number of members increases, the member length decreases. Effectively, in the second set of curves, the total weight of the hoop remains fixed. Deployment time is measured from the time the packaged hoop is released to the time all the joints lock up.





DYNAMIC ANALYSIS OF SPACE STRUCTURES INCLUDING ELASTIC, MULTIBODY, AND CONTROL BEHAVIOR

Larry Pinson
Langley Research Center
Hampton, Virginia

Keto Soosaar
Cambridge Research

The Problem

To develop analysis methods, modeling strategies, and simulation tools to predict with assurance the on-orbit performance and integrity of large complex space structures that cannot be verified on the ground.

Problem Incorporates:

- Large Reliable Structural Models (including non-linear)
- Multi-Body Flexible Dynamics
- Multi-Tier Controller Interaction
- Environmental Models Including 1g and Atmosphere
- Various On-Board Disturbances
- Linkage to Mission-Level Performance Codes

All areas are in serious need of work, but weakest link is multi-body flexible dynamics.

Some Definitions

Structural Dynamics: Motions of an elastic continuous structure under time-varying forces.

Dynamics: Motions of a rigid particle or continuum.

Multi-Body Dynamics: Motions of an assembly of rigid and/or flexible elements mutually interacting via non-elastic connections (trees or rings)

Multi-Body Dynamics are Encounted in Spacecraft with:

1. Very Flexible Fixed Appendages
2. Rotating Appendages
3. Dual-Spinners
4. Isolators or Gimbals between Significant Parts of S/C
5. During Deployments

MULTI-BODY TOOLS WILL PROBABLY BE NEEDED FOR:

NASA SSTM	NAME
A-18	PINHOLE OCCULTER FACILITY (50 M)
A-20	LARGE DEPLOYABLE REFLECTOR (20 M)
C-6	GEOSTATIONARY PLATFORM
U-4	TETHERED SATELLITE
U-5	SPACE STATION
A-24	INFRARED RADIOMETER (100 M)
A-25	GRAVITY WAVE INTERFEROMETER (1,000 M)
A-26	COSMIC (34 M)
A-27	100 M THINNED APERTURE
A-28	VERY LARGE SPACE TELESCOPE
L-1	SEARCH FOR EXTRA-TERRESTRIAL INTELLIGENCE (300 M)
U-6	GEOSYNCHRONOUS SPACE STATION

Multi-Body Dynamics Code Needs can be Gathered into Following Classes:

1. Large Area Antenna
2. Space Station
3. Generalized Deployment
4. Optical Systems
5. Miscellaneous General-Purpose Codes

GENERAL-PURPOSE CODE

- FIRST-ORDER ASSESSMENT OF NEW CONCEPTS
  - SAILS, TETHERS, MULTI-RINGS, DEPLOYMENTS
- SMALL TO MEDIUM-SIZE PROBLEMS
- CONTROL-STRUCTURE INTERACTION
- LARGE MINI-COMPUTER ENVIRONMENT, MACHINE INDEPENDENT
- USER-FRIENDLY, FLEXIBLE
- EVOLUTIONARY VERSION OF CURRENT DISCOS

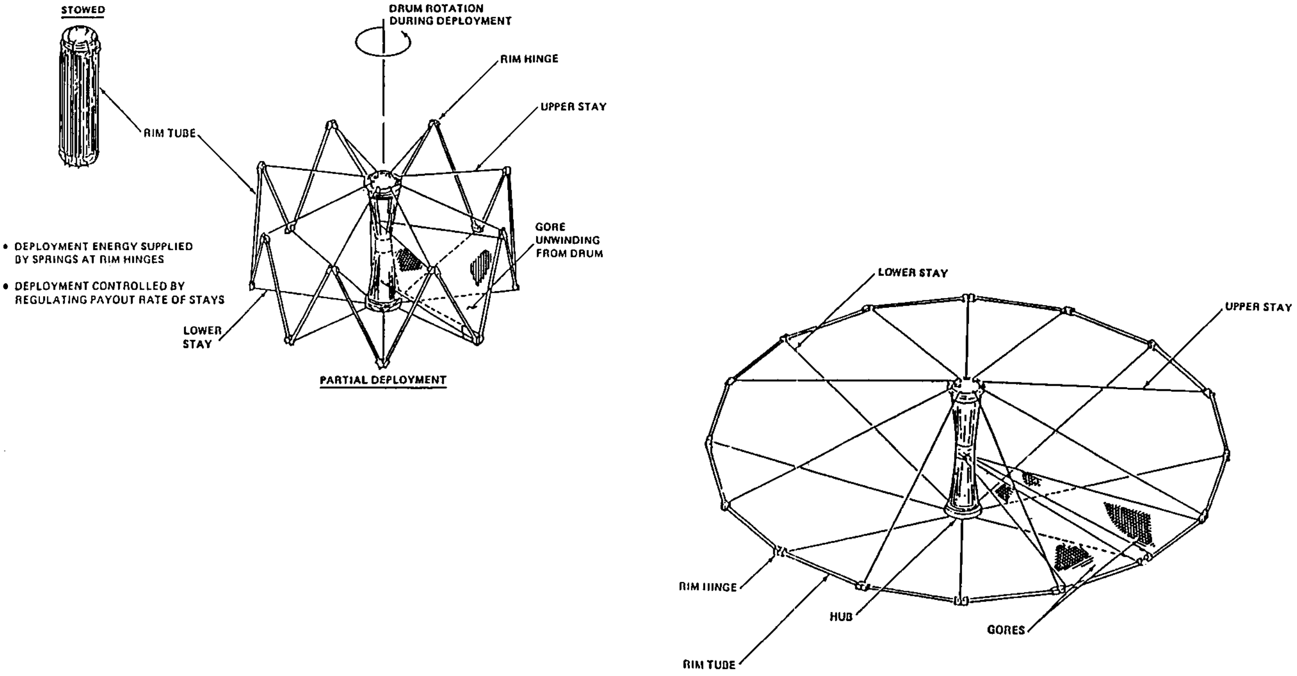
DEPLOYMENT CODE

- DRIVEN MAINLY BY LARGE LIGHTWEIGHT ANTENNAS
- TREES OR RINGS WITH MANY BODIES
- MASS FLOW DURING DEPLOYMENT
- GEOMETRIC STRUCTURAL NON-LINEARITIES
- TIME-VARYING LARGE STRUCTURAL MODEL
- OPEN OR CLOSED-LOOP CONTROL OF DEPLOYMENT

ASSESSMENT ISSUES

- DEPLOYMENT INTO UNACCEPTABLE CONFIGURATION
- DEPLOYMENT INTO NON-RECOVERABLE SPIN MODES
- ENTANGLEMENTS, BREAKAGE, STRUCTURAL INSTABILITY

LARGE ANTENNA
DEPLOYMENT


STOWED
RIM TUBE
DRUM ROTATION DURING DEPLOYMENT
RIM HINGE
UPPER STAY
GORE UNWINDING FROM DRUM
• DEPLOYMENT ENERGY SUPPLIED BY SPRINGS AT RIM HINGES
• DEPLOYMENT CONTROLLED BY REGULATING PAYOUT RATE OF STAYS
LOWER STAY
PARTIAL DEPLOYMENT
LOWER STAY
UPPER STAY
RIM HINGE
HUB
GORES
RIM TUBE

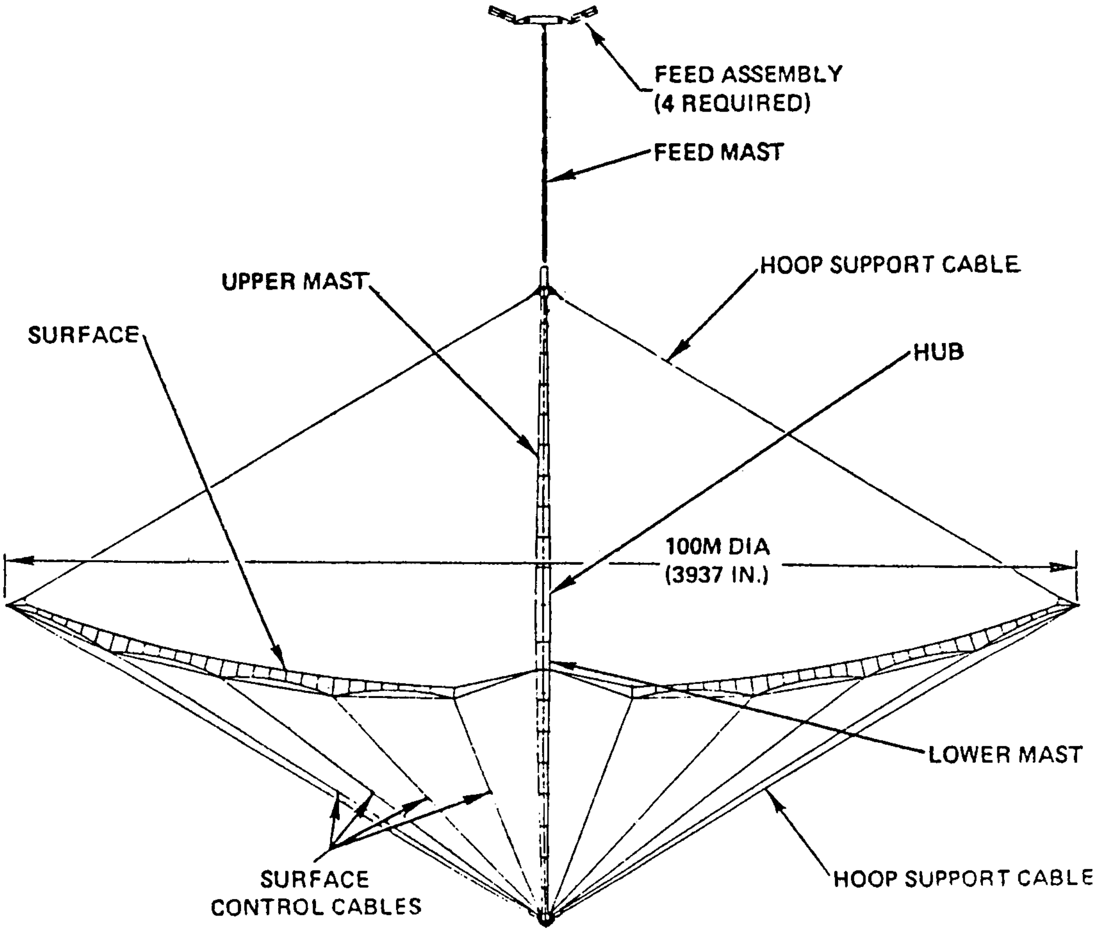
VERY LARGE ANTENNA CODE

- OPERATIONAL CONFIGURATION - LIMITED MULTI-BODY
- VERY LOW-FREQUENCY STRUCTURE
- VERY LARGE STRUCTURAL MODEL (10-50,000 DOF)
- MEMBRANE OR OTHER GEOMETRIC NONLINEARITIES
- CONTROLLED SURFACE, FEED ALIGNMENT, SYSTEM POINTING
- MODAL VS. TRAVELLING-WAVE REPRESENTATION

ASSESSMENT ISSUES

- MAIN LOBE LOSS OF GAIN
- SIDE-LOBE STRUCTURE
- DYNAMIC INTERACTION WITH ENVIRONMENTAL DISTURBANCES
- MAJOR STRUCTURE-CONTROL INTERACTION

TYPICAL
LARGE ANTENNA



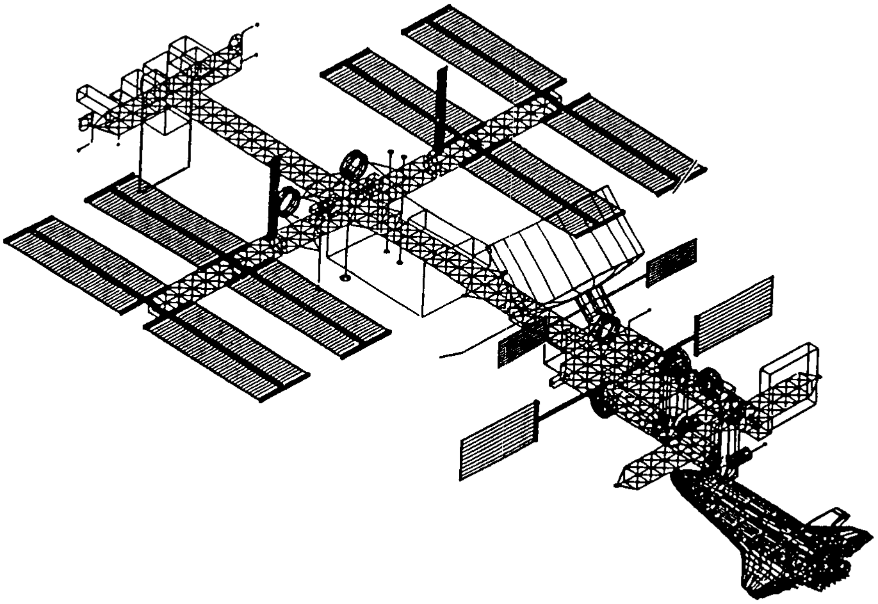
SPACE STATION CODE

- MULTI-BODY TREES (APPENDAGES & PAYLOAD SENSORS)
- LARGE STRUCTURAL MODEL
- SYSTEM AND EXPERIMENT POINTING CONTROL
- SIGNIFICANT INERTIA CHANGES (CONSTRUCTION, DOCKING)
- EXPERIMENT DISTURBANCES

ASSESSMENT ISSUES

- EXPERIMENT ISOLATION FROM ACCELERATION
- EXPERIMENT POINTING & TRACKING
- OCCUPANT COMFORT
- CONSUMABLES

SPACE STATION



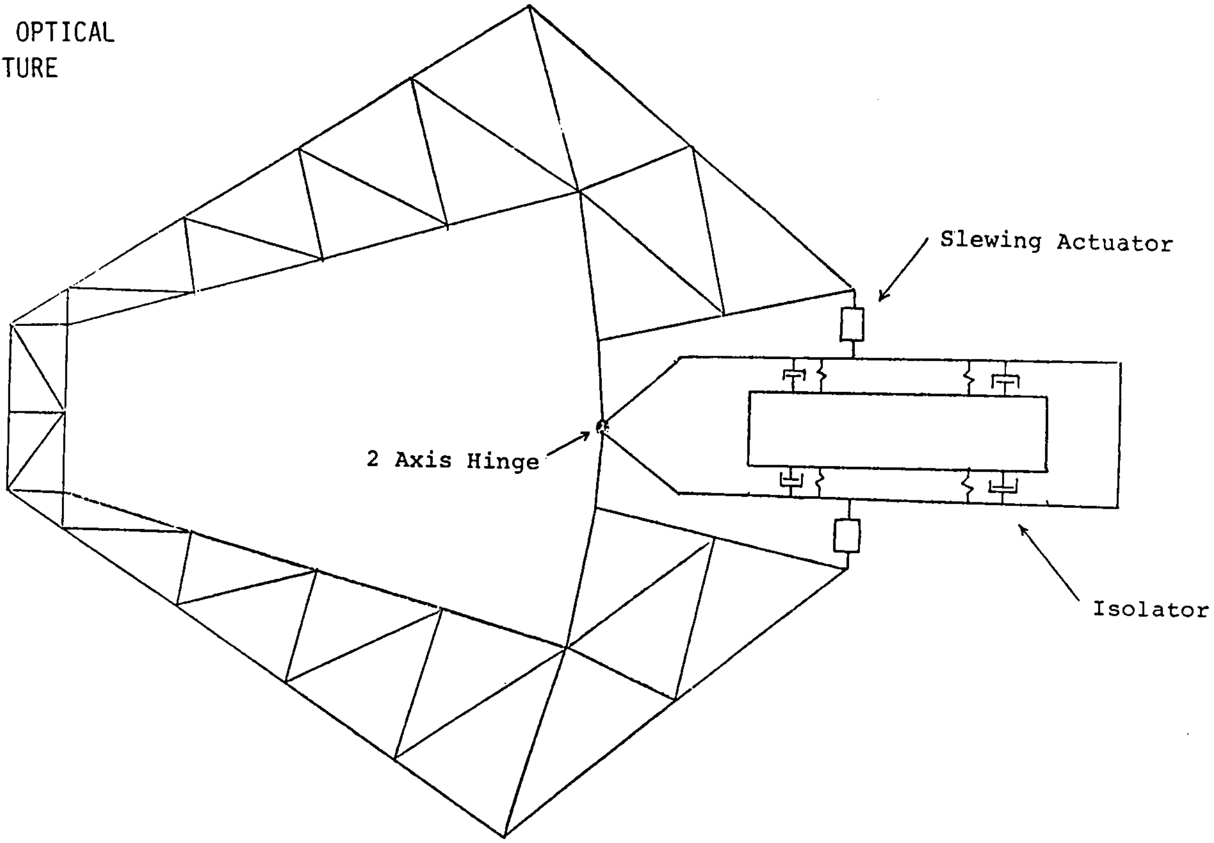
OPTICAL STRUCTURES CODE

- OVERLAPPING CONTROL SYSTEMS
  - SURFACE (WAVEFRONT)
  - VIBRATION
  - RAPID SLEW
  - PRECISION POINTING
- MULTIBODY (TREES)
- ISOLATORS
- MANY SOURCES OF DISTURBANCE
- SLOSH AND POGO
- RAPIDLY VARYING INERTIAS
- RAPID CONFIGURATIONAL CHANGES
- VERY LARGE ELASTIC MODEL

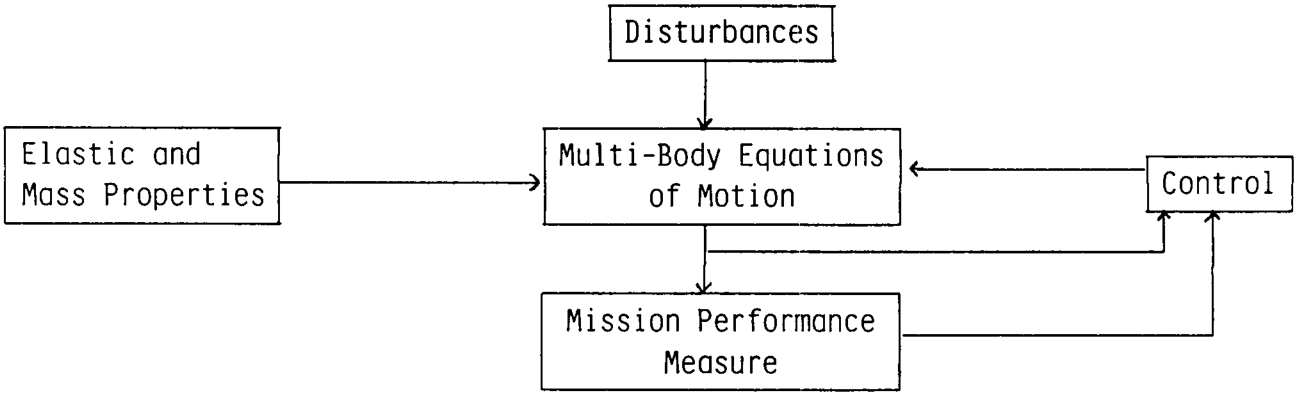
ASSESSMENT ISSUES

- SYSTEMS-LEVEL PERFORMANCE (LINKAGE TO OPTICS CODE)
- ROBUSTNESS OF MULTI-TIER CONTROL

TYPICAL OPTICAL STRUCTURE



Generic Assessment Tool


Disturbances
Elastic and Mass Properties
Multi-Body Equations of Motion
Control
Mission Performance Measure

STATUS OF SPACE-SYSTEMS ORIENTED MULTI-BODY TECHNOLOGY

- DIVERSITY OF FORMULATIONS
  - TWO GENERAL FAMILIES
    - ANALYTICAL MECHANICS - "DISPLACEMENT METHOD"
    - EULER/NEWTON - "FORCE METHOD"
  - SEVERAL SCHOOLS OF THOUGHT WITHIN FAMILIES
- DIVERSITY OF SOFTWARE CODES
  - SOME EXCELLENT, MANY MARGINAL
  - SIGNIFICANT LEARNING CURVES, USER HOSTILE
  - GENERALLY LONG RUNNING TIMES
  - UNCERTAIN ACCURACY/VALIDITY
  - MANY USERS UNSOPHISTICATED, TREAT AS BLACK BOX
- GENERALLY AN IMMATURE AREA (UNLIKE STRUCTURAL DYNAMICS)

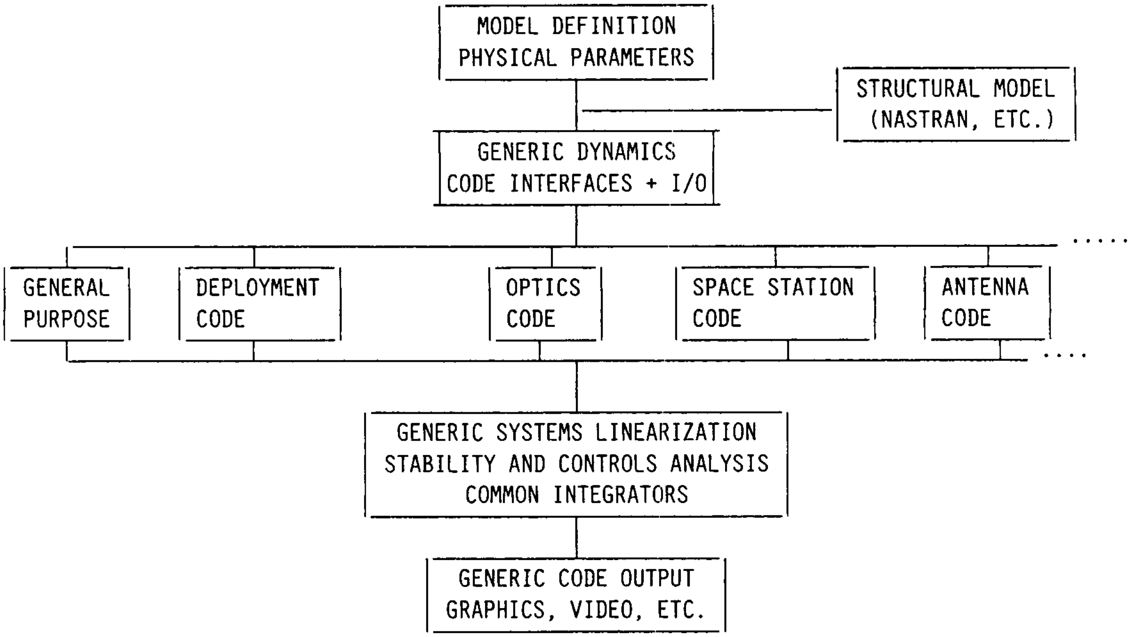
CONCERN:

- We are proposing more complicated satellites than our current analytical tools can reliably predict.
- In the multi-body area there is a vast diversity of opinion on the proper approach to the formulations.
- The time to develop a unified formulation, and convert it into code, will exceed the time available for immediate needs.

Two Approaches to Resolution

- Integration of available and other near-term codes (2-4 years).
- Basic research and development activity leading to NASTRAN-like multi-body code (5-8 years).

PATCHED DYNAMICS CODES (INTERIM SOLUTION)



MODEL DEFINITION PHYSICAL PARAMETERS

STRUCTURAL MODEL (NASTRAN, ETC.)

GENERIC DYNAMICS CODE INTERFACES + I/O

GENERAL PURPOSE

DEPLOYMENT CODE

OPTICS CODE

SPACE STATION CODE

ANTENNA CODE

GENERIC SYSTEMS LINEARIZATION STABILITY AND CONTROLS ANALYSIS COMMON INTEGRATORS

GENERIC CODE OUTPUT GRAPHICS, VIDEO, ETC.

OBJECTIVES OF NEW MULTI-USER CODE

- ENDURING BUT EFFICIENT COMMON FORMULATION
  - TREES, RINGS, MASSFLOW
  - LARGE STRUCTURAL MODELS
  - MULTI-LEVEL CONTROL

- SOFTWARE FEATURES
  - USER-FRIENDLY PROBLEM-LANGUAGE I-O
  - OBJECT-ORIENTED PROBLEM ASSEMBLY
  - INCORPORATED SYMBOLIC MANIPULATION
  - STRIPPED, EFFICIENT CODE FOR EXECUTION

- MACHINE-INDEPENDENCE AND ACCESSIBILITY
  - SUPER-MINIS
  - MAINFRAMES
  - SUPERS
  - FEDERATED PARALLEL PROCESSORS

Basic Approach to Development

- Consolidate Multi-Agency Government Support
- Theory Phase $T = T_0$
  - Technical Participation by Government, Industry, Academia
  - Study and Consolidation of Alternate Formulations
  - Preliminary Software Architecture Studies
- Prototype Phase $T = T_0 + 2$
  - Reduce to 2 or 3 Major Formulation and Software Approaches
  - Continue Support to Universities to Train Users
- Coding Phase $T = T_0 + 3$
  - Choose Best Overall Approach to Code
- Preliminary Testing Phase $T = T_0 + 5$
  - First Release to Selected Users
- Public Release $T = T_0 + 6$

Summary

- The problems are there, funding should be pursued
- On-going capabilities fall short
- Near-term needs require the integration of existing codes
- Far-term needs must follow a return to basics

Part I

DYNAMICS OF FLEXIBLE MULTI-BODY MECHANISMS AND MANIPULATORS

An Overview

Steven Dubowsky
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA

INTRODUCTION

A. FLEXIBILITY CAN BE A MAJOR LIMITATION TO THE PERFORMANCE OF HIGH PERFORMANCE CONVENTIONAL MACHINE SYSTEMS.

1. NOISE
2. VIBRATION
3. WEAR
4. PREMATURE FAILURES
5. DESTABILIZE CONTROL

B. THE CURRENT STATE-OF-THE-ART OF ROBOTIC MANIPULATORS IS LIMITED BY THE EFFECTS OF SYSTEM FLEXIBILITY.

THE STATE-OF-THE-ART OF THE ROBOT DYNAMICS AND CONTROL[†]

*NOW:
(CURRENT COMMERCIAL SYSTEMS)

*IN 5 YEARS
(LABORATORY DEMONSTRATED)

*IN 10 YEARS
(CURRENT RESEARCH ISSUES)

[†] This chart defines the time frames for the review of the state of the art for robotic systems which follow and provide the basis for the future projections

NOW

1. ROBOTS ARE:
   - * Not Robots
   - * Individual Arms on Fixed Bases, or
   - * Simple Guided Vehicles

2. MECHANICAL DESIGN :
   - * Heavy, Rigid and Slow

3. SENSORS :
   - * Simple Joint Transducers
   - * Primitive 2-D Vision
   - * Rudimentary Force Sensors

4. ACTUATORS :
   - * Heavy and Low Power
   - * Troublesome Transmissions

5. END EFFECTORS :
   - * Binary
   - * With Simple Sensors
   - * Special Purpose Tools

6. MOTIONS :
   - * Not Dynamic - "Quasi-Static"
   - * Speeds Below Structural Resonances

7. CONTROL :
   - * Primitive Linear Joint Control
   - * Low Performance
   - * No Absolute Position Accuracy
   - * Only Static Force Control
   - * No Dynamic Trajectory Planning

IN 5 YEARS

1. ROBOTS ARE:
    * Still Not Robots
    * 2 or 3 Fixed Arms Working Together
    * Some Mobility

2. MECHANICAL DESIGNS :
    * Rigid, Light and Faster

3. SENSORS :
    * Still Mostly Joint Transducers
    * Some VLSI 2-D Vision
    * Simple End-Point Sensors

4. ACTUATORS :
    * Lighter Weight and Improved
    * Direct Drives

5. END EFFECTORS :
    * Some Controlled Mobility
    * Position, Force and Limited Tactile Sensing
    * Commercial Tools for Some Tasks

6. MOTIONS :
    * Control Permits "Dynamic" Performance
    * Speeds Below Structural Resonances

7. CONTROL :
    * Combined Position and Force
    * "Work-Space" Rather Than of the Joints
    * Insensitive to Environmental Changes
    * Optimal Dynamic Trajectory Planning

IN 10 YEARS

1. ROBOTS MAYBE :
   * Robots
   * Coordinated Multiple and Mobile Arms
   * Self-Contained with Walking Ability

2. MECHANICAL DESIGNS :
   * Very Light, flexible and fast

3. SENSORS :
   * New Sensor Technologies for Control
   * High Speed 3-D Vision
   * High Resolution Tactile Sensors

4. ACTUATORS :
   * High Peformance
   * New Technologies - Muscle Types

5. END EFFECTORS :
   * Sensitive and Dexterous Hands
   * Intelligent Motion and Sensing
   * Intelligent Tools for Specific Tasks

6. MOTIONS :
   * Dynamically Tuned
   * Flexibility Exploited for Performance

7. CONTROL :
   * Issues of Control and Performance in Most Cases Will Move to a Higher Level.
   * Questions of Control of Individual Robot Actions Will be Transparent.

N89-24663

Part II

APPLICATION OF FINITE-ELEMENT METHODS TO DYNAMIC ANALYSIS OF FLEXIBLE SPATIAL AND CO-PLANAR LINKAGE SYSTEMS

Steven Dubowsky

The following figures describe an approach to modeling the flexibility effects in spatial mechanisms and manipulator systems. The method is based on finite-element representations of the individual links in the system. However, it should be noted that conventional FEM methods and software packages will not handle the highly nonlinear dynamic behavior of these systems which result from their changing geometry. In order to design high-performance lightweight systems and their control systems, good models of their dynamic behavior which include the effects of flexibility are required.

FOCUS

- **DEVELOP PRACTICAL AND EFFICIENT METHODS WHICH ANALYZE SPATIAL MECHANISMS AND MANIPULATORS CONTAINING IRREGULARLY SHAPED FLEXIBLE LINKS**

The method presented here for the modeling of the dynamic behavior of manipulators and machine systems with flexibility is based on using individual finite-element link models to reduce the number of dynamic degrees of freedom. The system gross motion is modeled using 4 by 4 matrix methods. The resulting equations of motion contain both the full nonlinear behavior introduced by the system's gross motion and the effects of link flexibility.

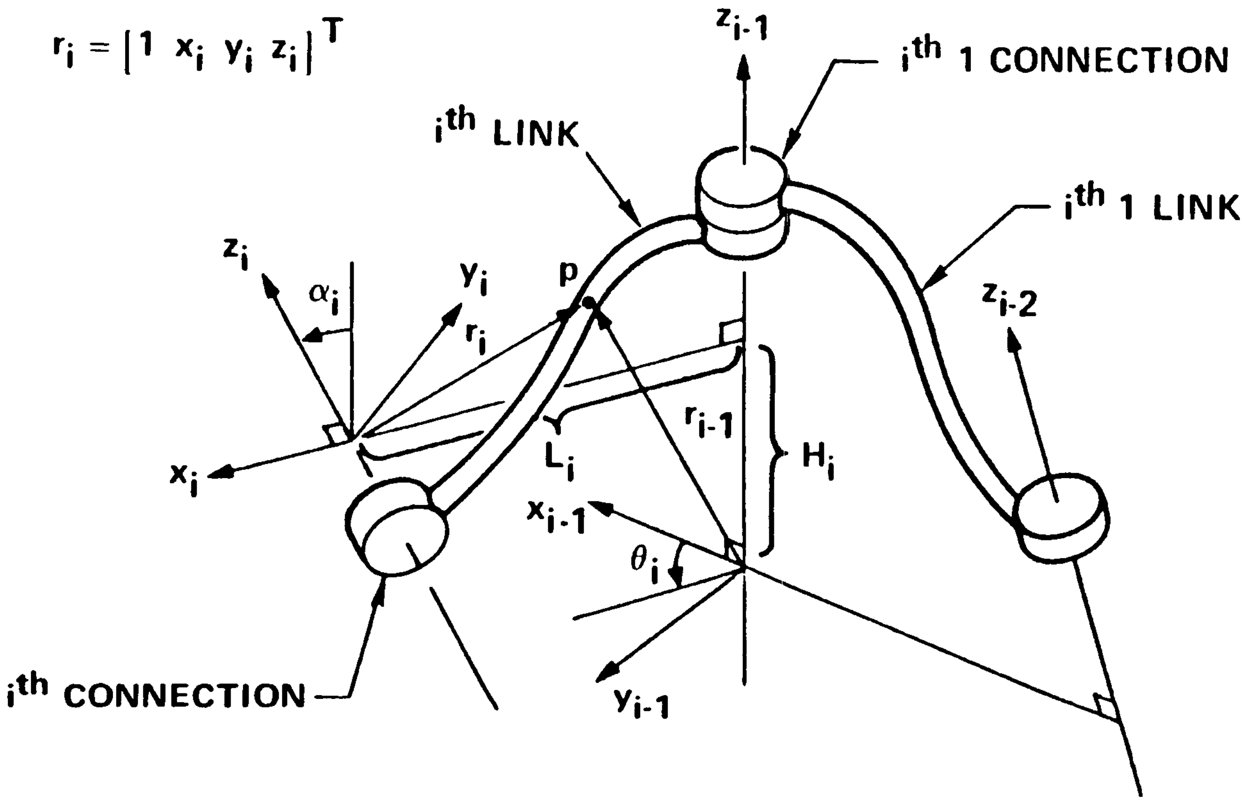
ANALYTICAL APPROACH

- 4 x 4 MATRIX DYNAMIC ANALYSIS TECHNIQUES
    - WELL-ESTABLISHED METHOD
    - APPLIED TO RIGID LINK SYSTEMS IN PREVIOUS WORK
    - POSSIBLE TO EXTEND ANALYSIS TO INCLUDE FLEXIBILITY OF LINKS
- FINITE-ELEMENT METHODOLOGY
    - USED EXTENSIVELY IN STRUCTURAL DYNAMICS
    - STANDARD FINITE-ELEMENT PROGRAMS (NASTRAN, SAP, ETC) ARE WIDELY AVAILABLE
- PERTURBATION COORDINATES
- COMPONENT MODE SYNTHESIS COORDINATE REDUCTION

This figure defines the well-known 4 by 4 coordinate transformation. These transformations contain the information that describes the kinematic constraints imposed by the systems joints or connections.

4 x 4 MATRIX NOTATION

$$r_i = [1 \ x_i \ y_i \ z_i]^T$$



The vectors representing any point in the system can be represented to a common frame using 4 by 4 methods. In particular, the inertial position of any point can be described.

4 x 4 MATRIX ANALYSIS

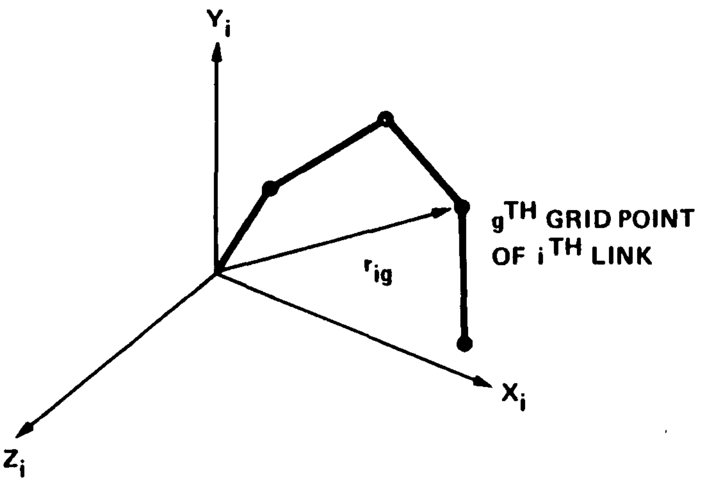
$$r_{i-1} = T^{i}_{i-1} r_i$$

$$T^{i}_{i-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ L_i \cos\theta_i & \cos\theta_i & -\sin\theta_i \ \cos a_i & \sin\theta_i \ \sin a_i \\ L_i \sin\theta_i & \sin\theta_i & \cos\theta_i \ \cos a_i & -\cos\theta_i \ \sin a_i \\ H_i & 0 & \sin a_i & \cos a_i \end{bmatrix}$$

$$\overset{0}{r_i} = T^{1}_{0} T^{2}_{1} T^{3}_{2} \ . \ . \ . \ T^{i}_{i-1} r_i = T^{i}_{0} r_i$$

The position variables of the finite-element grid points must be transformed into 4 by 4 notation.

LOCAL GRID POINT MOTION



LOCAL POSITION:

$$r_{ig} = \sum_{\beta=1}^{NP(i)} \Phi_{ig\beta}\, p_{i\beta} + b_{ig}$$

NOMINAL POSITION:

$$b_{ig} = [1 \; x_{ig} \; y_{ig} \; z_{ig}]^T$$

SELECTION VECTOR:

$$\begin{aligned}\Phi_{ig\beta} &= [0\,1\,0\,0]^T \text{ for } \beta = 1 + 6(g\text{-}1)\\ &= [0\,0\,1\,0]^T \text{ for } \beta = 2 + 6(g\text{-}1)\\ &= [0\,0\,0\,1]^T \text{ for } \beta = 3 + 6(g\text{-}1)\\ &= [0\,0\,0\,0]^T \text{ for all other } \beta\end{aligned}$$

The inertial velocities of the grid points are calculated in 4 by 4 notation so that the kinetic energy (next figure) required by Lagrange's Equations (following figure) can be formulated.

GRID POINT INERTIAL VELOCITY

- INERTIAL POSITION:

$$r^{o}_{ig} = T^{i}_{o}\, r_{ig}$$

- INERTIAL VELOCITY:

$$v_{ig} = \sum_{j=1}^{i} U_{ij}\, r_{ig}\, \dot{\theta}_{j} + T^{i}_{o}\, \dot{r}_{ig}$$

$$\text{WHERE} \quad U_{ij} = \frac{\partial T^{i}_{o}}{\partial \theta'_{j}}$$

LINK ENERGY

- KINETIC ENERGY

$$T_i = \sum_{g=1}^{NG(i)} T_{ig} = \sum_{g=1}^{NG(i)} \frac{1}{2} m_{ig} \, Tr(v_{ig} \, v_{ig}^T)$$

- POTENTIAL ENERGY (ELASTIC)

$$V_i = \frac{1}{2} p_i^T \left[\widetilde{k_i}\right] p_i = \sum_{\beta=1}^{NP(i)} \sum_{\gamma=1}^{NP(i)} \widetilde{k}_{i\beta\gamma} \, p_{i\beta} \, p_{i\gamma}$$

LINK DYNAMIC EQUATIONS

- LAGRANGE'S EQUATIONS

$$\frac{d}{dt}\left[\frac{\partial(T_i)}{\partial \dot{p}_{ia}}\right] - \frac{\partial(T_i)}{\partial p_{ia}} + \frac{\partial(V_i)}{\partial p_{ia}} = \widetilde{f}_{ia} \qquad a = 1, \ldots, NP(i)$$

- LINK DYNAMIC EQUATIONS

$$m_i \ddot{p}_i + g_i \dot{p}_i + k_i p_i = f_i$$

The number of degrees of freedom for each link is reduced using component mode synthesis in order to achieve good computational efficiency.

COMPONENT MODE SYNTHESIS

- CMS TRANSFORMATION

$$p_i = \left[\begin{array}{c} p_i^I \\ \hline p_i^F \end{array}\right] = \left[\begin{array}{c|c} I & 0 \\ \hline \Phi_c & \Phi \end{array}\right] \left[\begin{array}{c} p_i^I \\ \hline \eta_i \end{array}\right] = A_i\, a_i$$

- REDUCED LINK DYNAMIC EQUATIONS

$$M_i \ddot{a}_i + G_i \dot{a}_i + K_i a_i = f_i^a$$

The link dynamic equations are formulated in terms of selected global coordinates.

GLOBAL EQUATIONS OF MOTION

- **GLOBAL TRANSFORMATION**

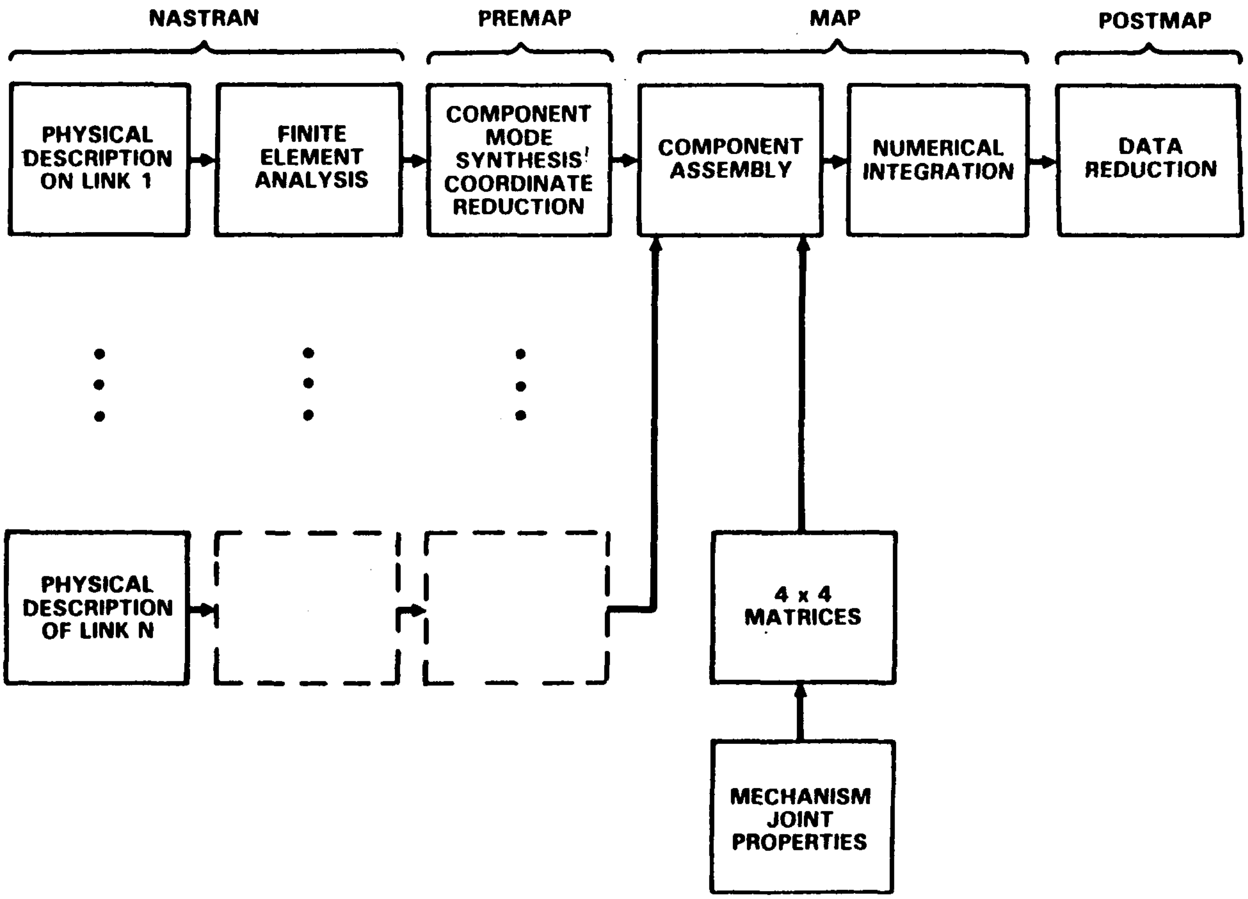
$$a_i = B_i\,(\theta\,(t))\;q$$

- **GLOBAL DYNAMIC EQUATIONS**

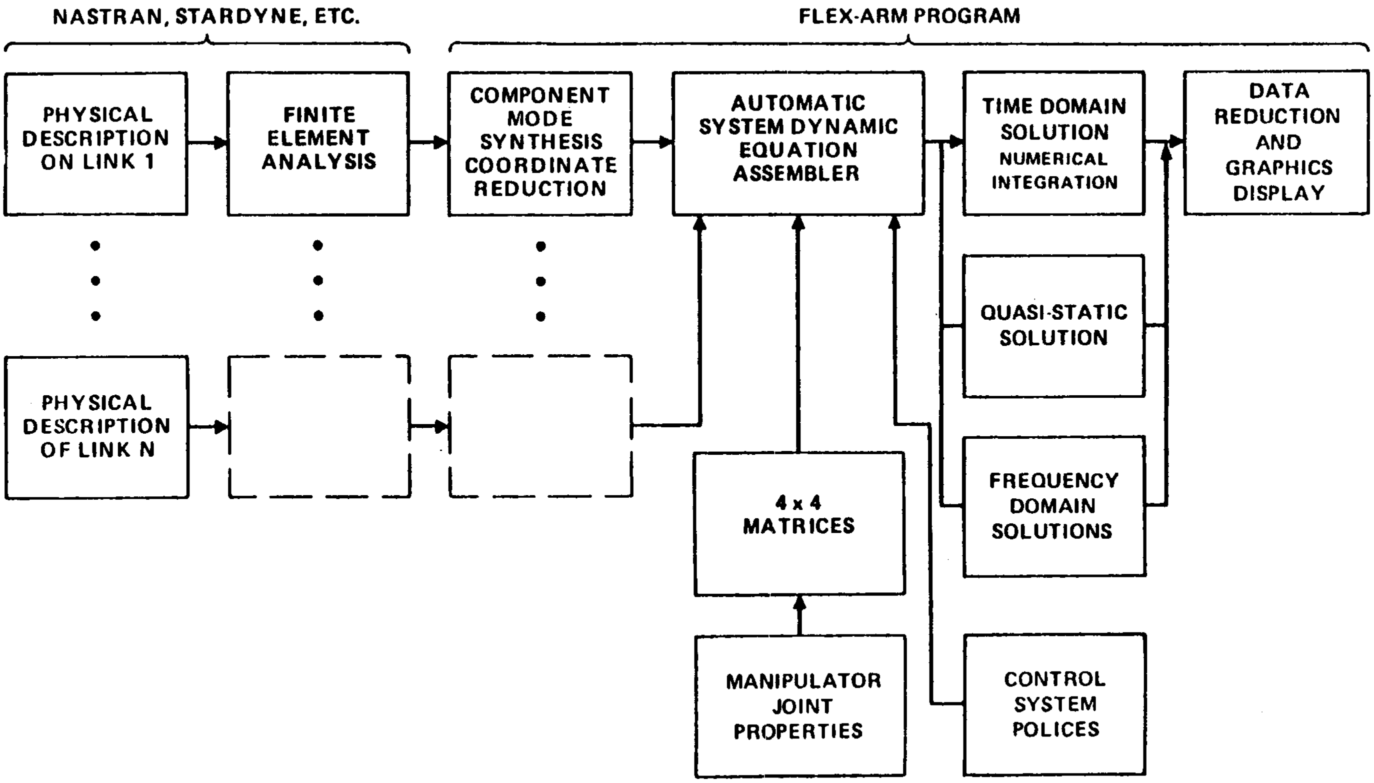
$$M\,(\theta)\,\ddot{q} + G\,(\theta,\dot{\theta})\,\dot{q} + K\,(\theta,\dot{\theta},\ddot{\theta})\,q = Q\,(\theta,\dot{\theta},\ddot{\theta},\,t)$$

This method has been automated in a software package called SALEM (Spatial Analysis of Linkages with Elastic Members).

SALEM ANALYSIS PROCEDURE


NASTRAN
PREMAP
MAP
POSTMAP
PHYSICAL DESCRIPTION ON LINK 1
FINITE ELEMENT ANALYSIS
COMPONENT MODE SYNTHESIS/ COORDINATE REDUCTION
COMPONENT ASSEMBLY
NUMERICAL INTEGRATION
DATA REDUCTION
PHYSICAL DESCRIPTION OF LINK N
4 x 4 MATRICES
MECHANISM JOINT PROPERTIES

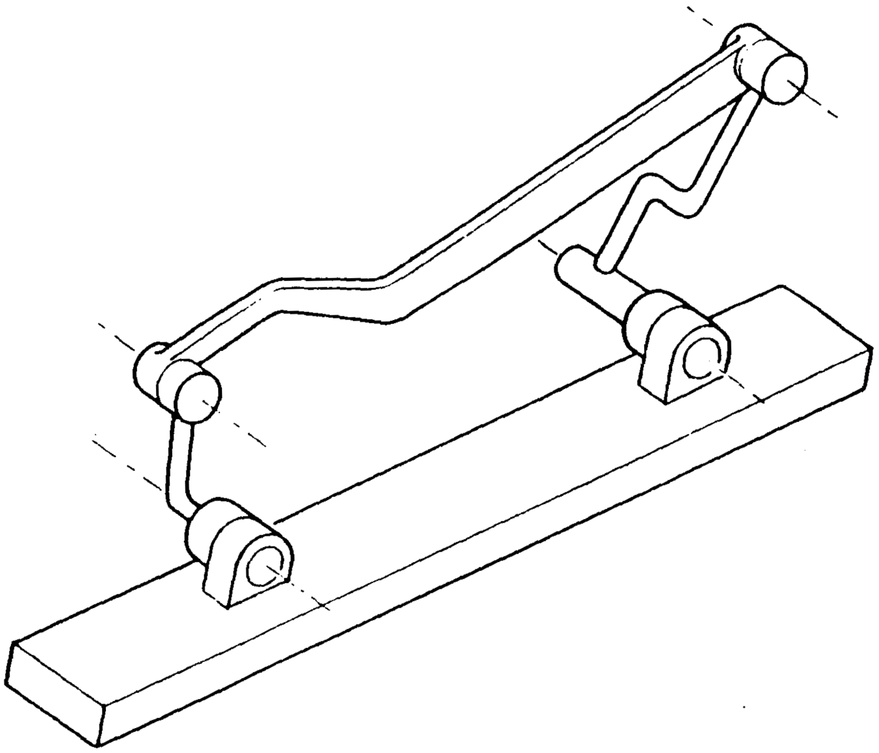
A special version, tailored for robotic manipulators, has also been created. This package is called FLEXARM (FLEXible Analysis of Robotic Manipulators). These programs include computer graphics output capabilities to assist the designer in visualizing, and hence, understanding the complex three-dimensional dynamic behavior of these systems. This figure shows the FLEXARM computational structure.


NASTRAN, STARDYNE, ETC.
FLEX-ARM PROGRAM
PHYSICAL DESCRIPTION ON LINK 1
FINITE ELEMENT ANALYSIS
COMPONENT MODE SYNTHESIS COORDINATE REDUCTION
AUTOMATIC SYSTEM DYNAMIC EQUATION ASSEMBLER
TIME DOMAIN SOLUTION NUMERICAL INTEGRATION
DATA REDUCTION AND GRAPHICS DISPLAY
QUASI-STATIC SOLUTION
PHYSICAL DESCRIPTION OF LINK N
4 x 4 MATRICES
FREQUENCY DOMAIN SOLUTIONS
MANIPULATOR JOINT PROPERTIES
CONTROL SYSTEM POLICES

Examples of the results which may be obtained using this technique are presented. First, a machine system will be considered and then results for a robotic manipulator will be presented.

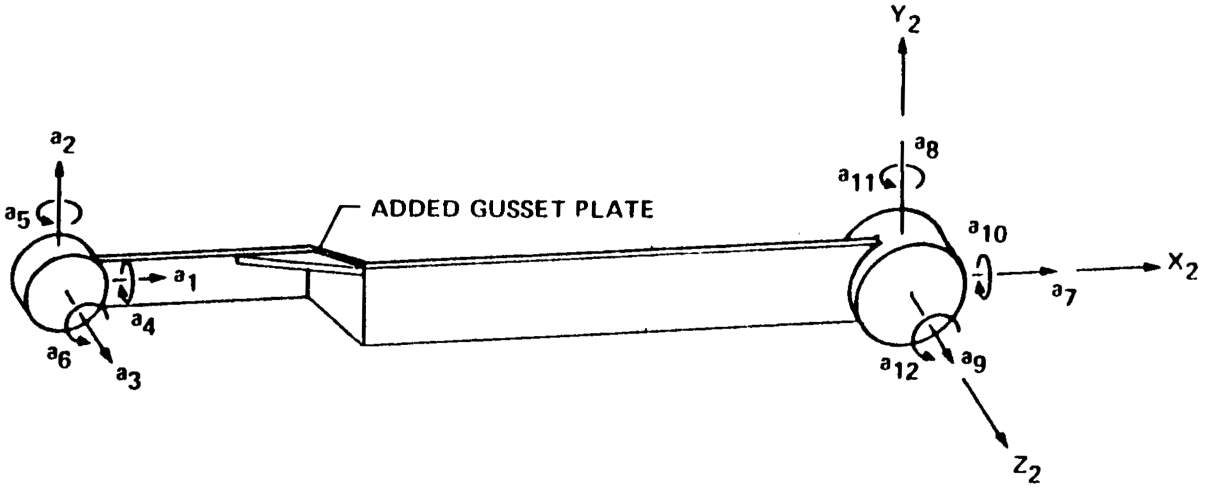
This figure shows a co-planar mechanism. Even though its kinematic structure is planar, it will exhibit spatial vibrations because of the off-sets in the links.

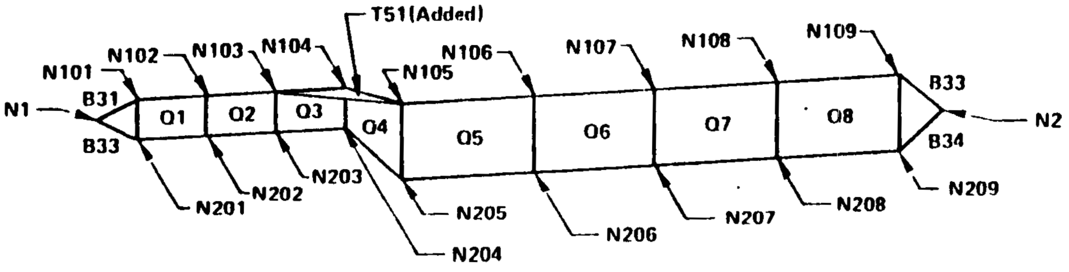
CO-PLANAR MECHANISM



This figure shows the details of the FEM model for the coupler link.

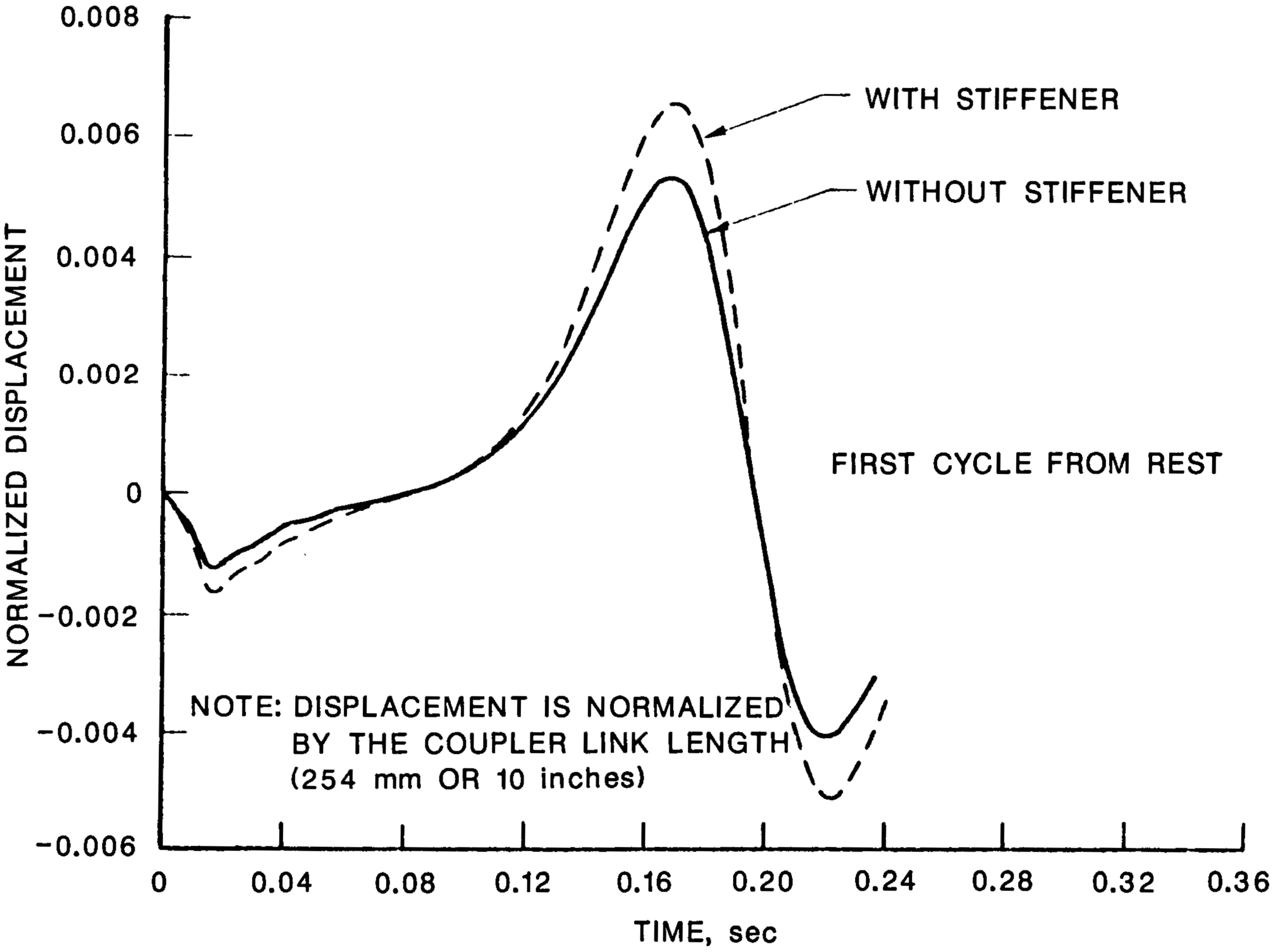
OFFSET COUPLER FINITE-ELEMENT MODEL





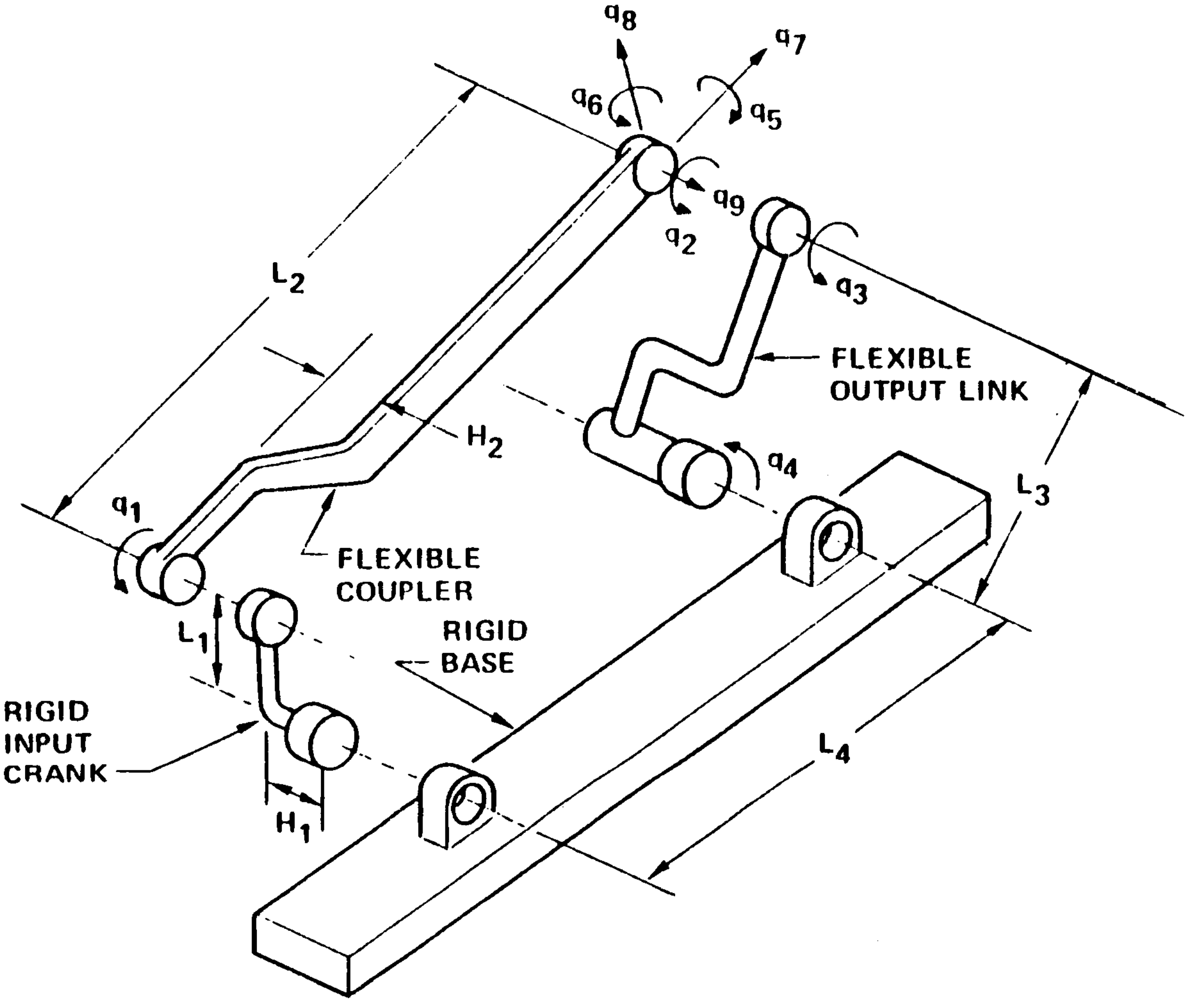
This is a typical plot of the displacements on the coupler link.

OUT-OF-PLANE DEFLECTION OF LINK



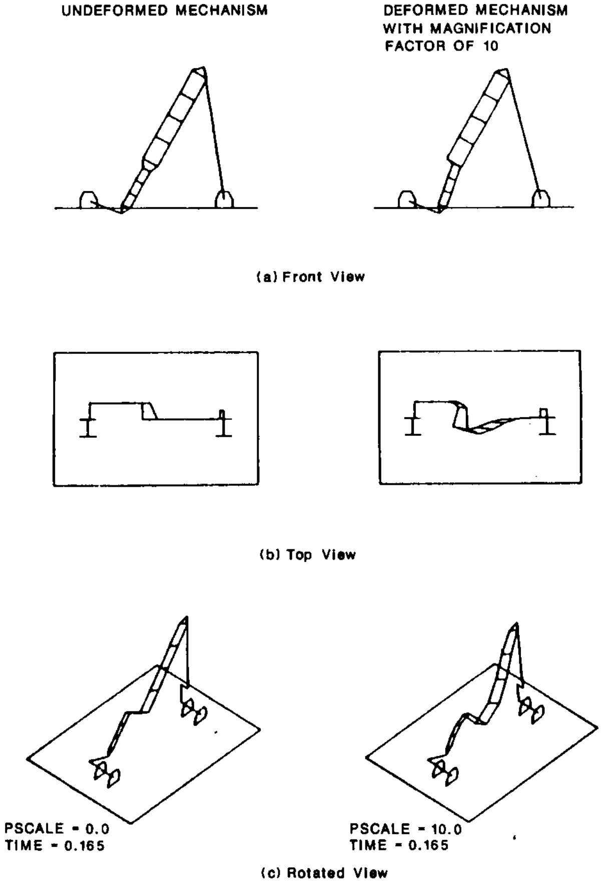
The global coordinates of the mechanism are presented here.

CO-PLANAR FOUR BAR LINKAGE



This figure shows different views of the deformation of the mechanism in one of its positions. This type of plot can be overlayed to create animated motions of the mechanisms motion.

CO-PLANAR FOUR-BAR DEFORMED GEOMETRY



(a) Front View

(b) Top View

(c) Rotated View

CONCLUSIONS

- A UNIFIED ANALYTICAL APPROACH FOR BOTH RIGID AND ELASTIC LINK MECHANISMS IS POSSIBLE

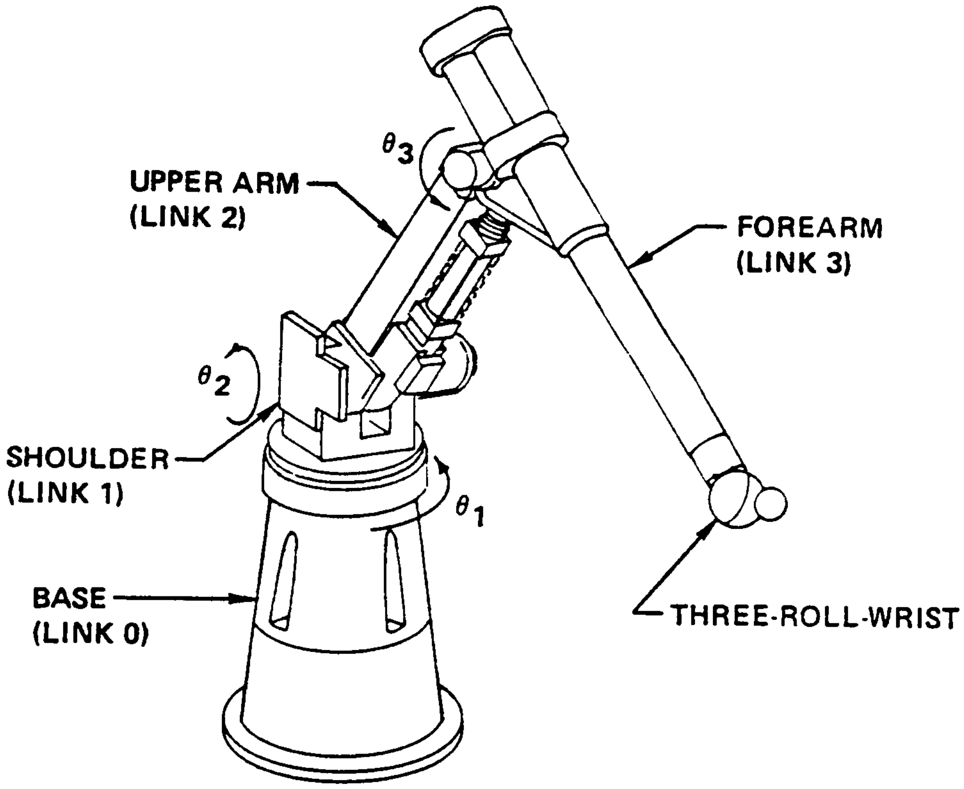
- EXISTING FINITE-ELEMENT PROCESSING PROGRAMS CAN BE FULLY UTILIZED TO REDUCE GEOMETRIC MODELING COMPLEXITY

- COMPONENT MODE SYNTHESIS COORDINATE REDUCTION IS IDEAL FOR USE IN FLEXIBLE LINKAGE ANALYSIS

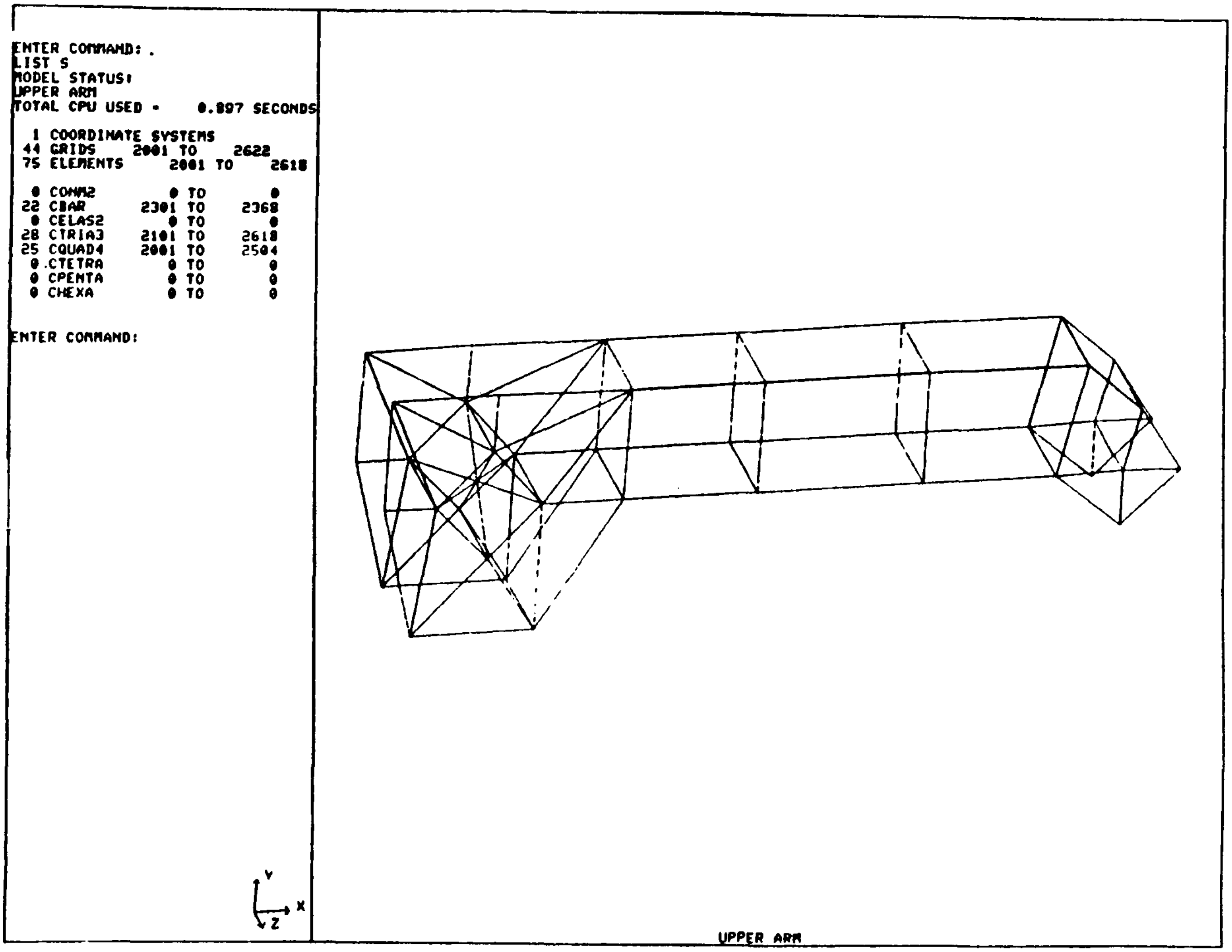
- INCREASED UNDERSTANDING OF 3D BEHAVIOR CAN BE OBTAINED THROUGH INTERACTIVE GRAPHICS

Part III

Shown below is an an example of the application of the method to a robotic manipulator: the Cincinnati MILACRON T3R3.



The first step in the method is to develop a standard NASTRAN FEM model for each link in the manipulator, including its base and the floor. The forearm model is shown below. The model includes such important parameters as the stiffness of the manipulators bearings.



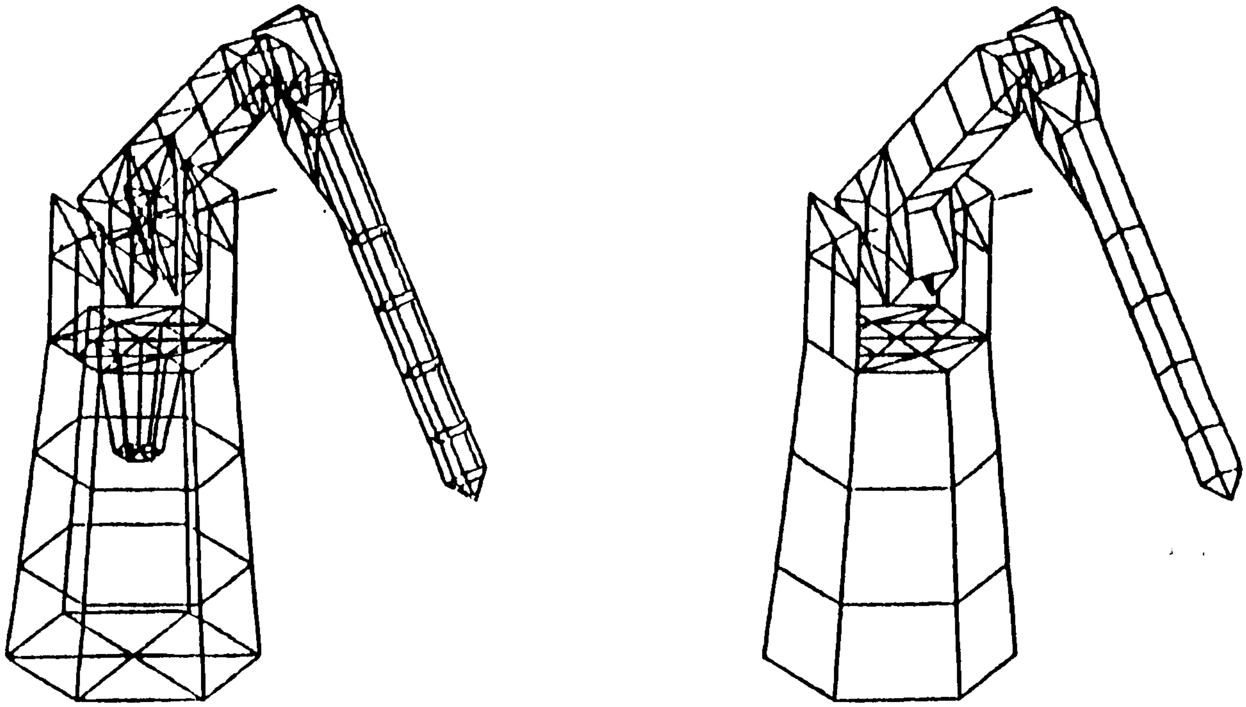
```
ENTER COMMAND: .
LIST S
MODEL STATUS:
UPPER ARM
TOTAL CPU USED =    0.897 SECONDS

  1 COORDINATE SYSTEMS
 44 GRIDS    2001 TO    2622
 75 ELEMENTS      2001 TO    2618

  0 CONM2          0 TO        0
 22 CBAR        2301 TO     2368
  0 CELAS2         0 TO        0
 28 CTRIA3      2101 TO     2618
 25 CQUAD4      2001 TO     2504
  0.CTETRA         0 TO        0
  0 CPENTA         0 TO        0
  0 CHEXA          0 TO        0

ENTER COMMAND:
```

In this figure the other NASTRAN models of the other links are shown. They have 155 grid points and 273 elements. With the degrees of freedom associated with the control systems, this unreduced system would have approximately 1650 DOF's. The computational cost required to simulate this large nonlinear system would be very high. However, the results obtained show that the structural degrees of freedom can be effectively reduced by CMS, and a total system model of less than 72 DOF's will yield high-quality results.

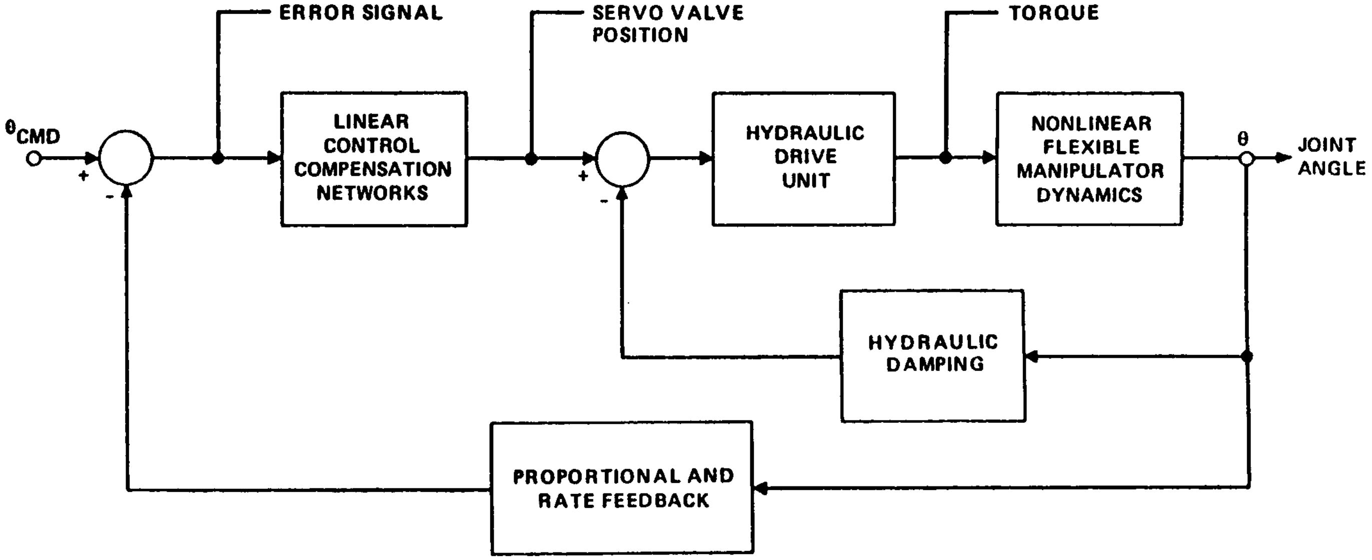


Complete Model

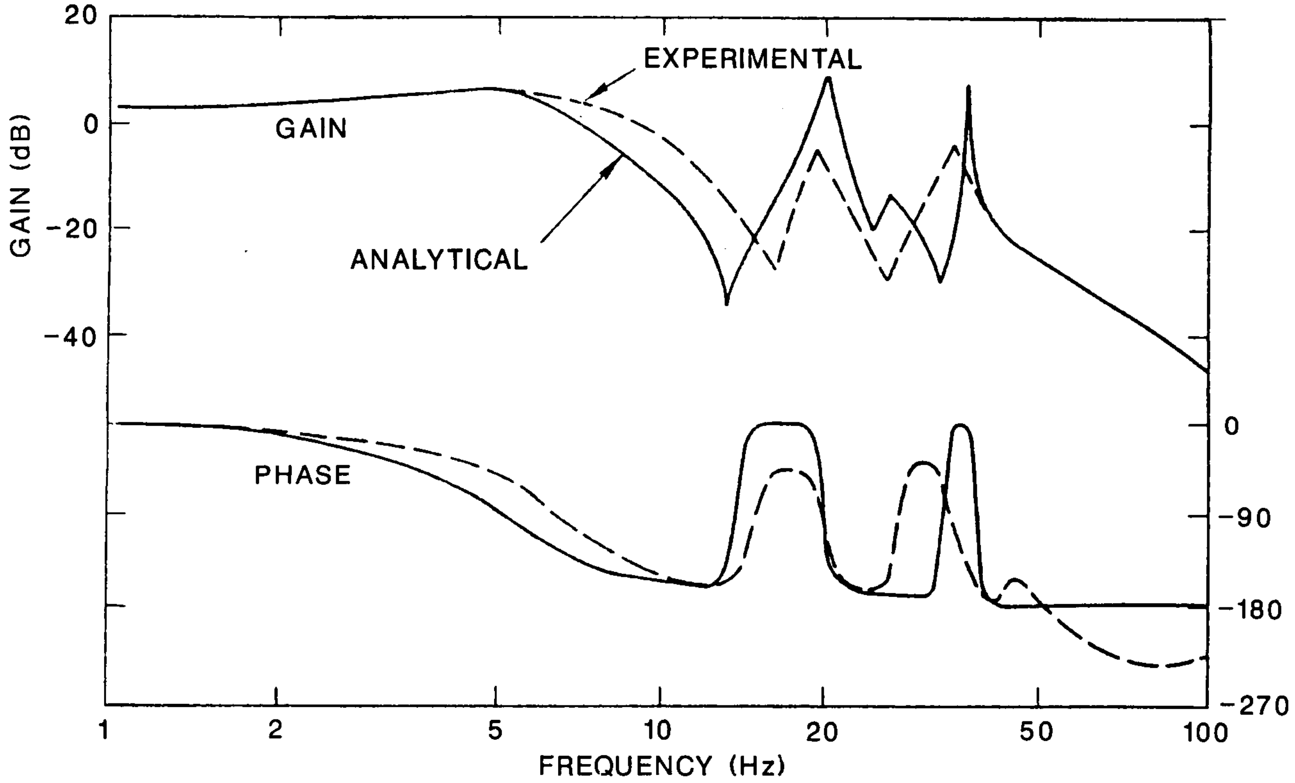
Model with Hidden Lines Removed

Detailed System NASTRAN Finite-element of Robotic Manipulator

An example of the typical control for one of the T3R3 axis is shown here.

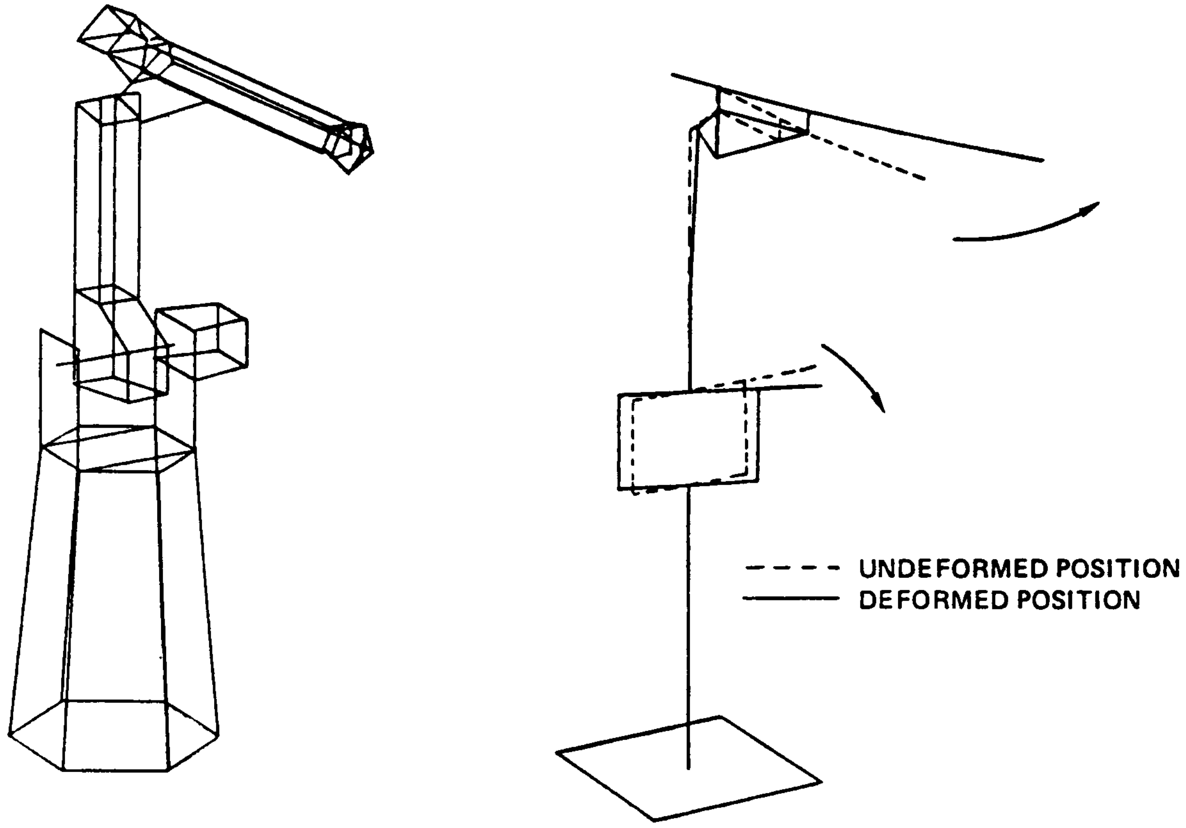


Frequency response experiments for the manipulator in a number of stationary positions show good agreement with the FLEXARM results. It should be noted that when the manipulator is nominally stationary, its equations are nearly linear and classical frequency response analysis is meaningful.



Comparison of Analytically and Experimentally Determined Serve Valve-to-Tachometer Transfer Functions for Base Drive

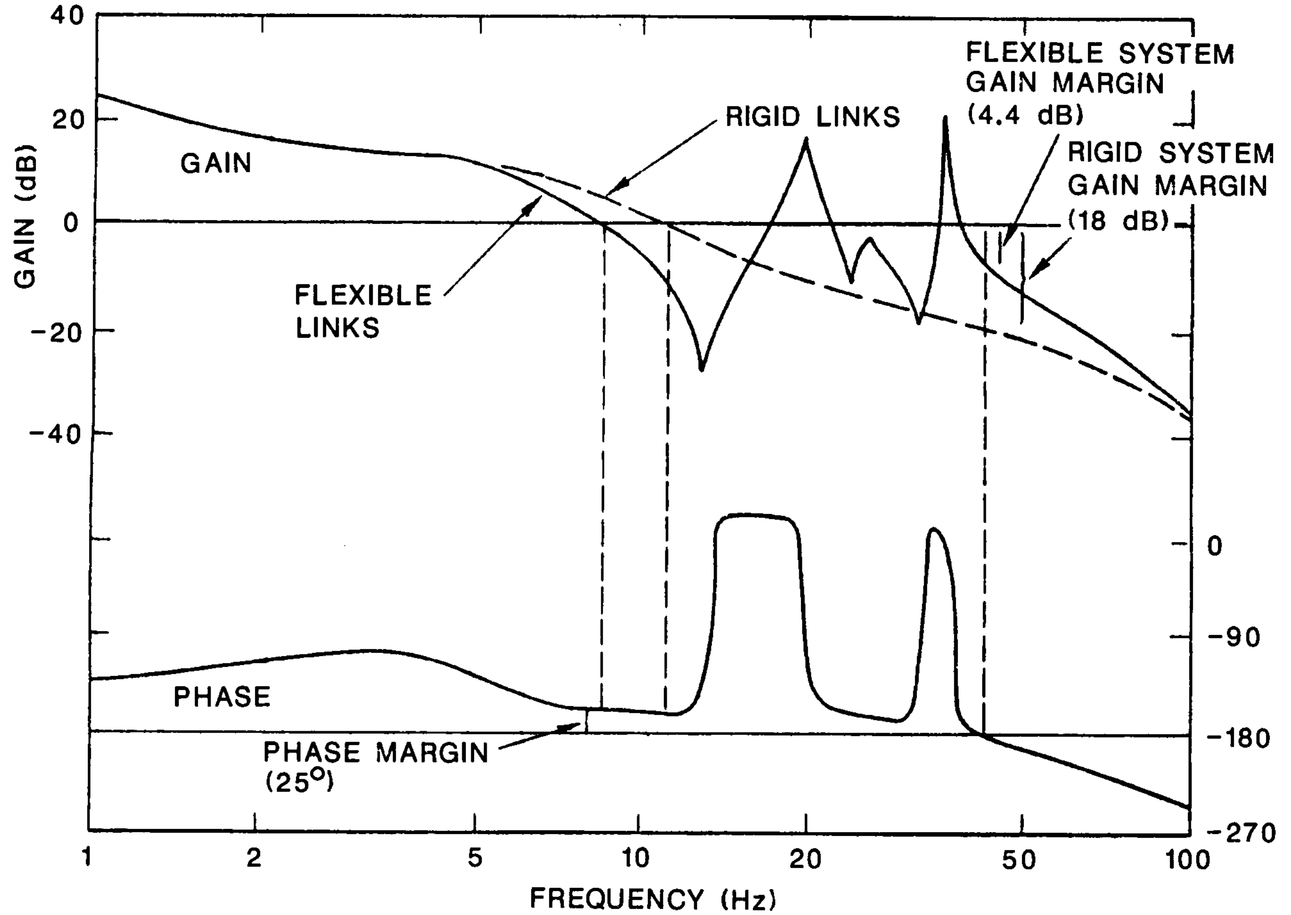
This figure shows an example of the manipulators first mode shape for a typical position obtained using FLEXARM.



a – TEST POSITION

b – MODE SHAPE AT 20 Hz

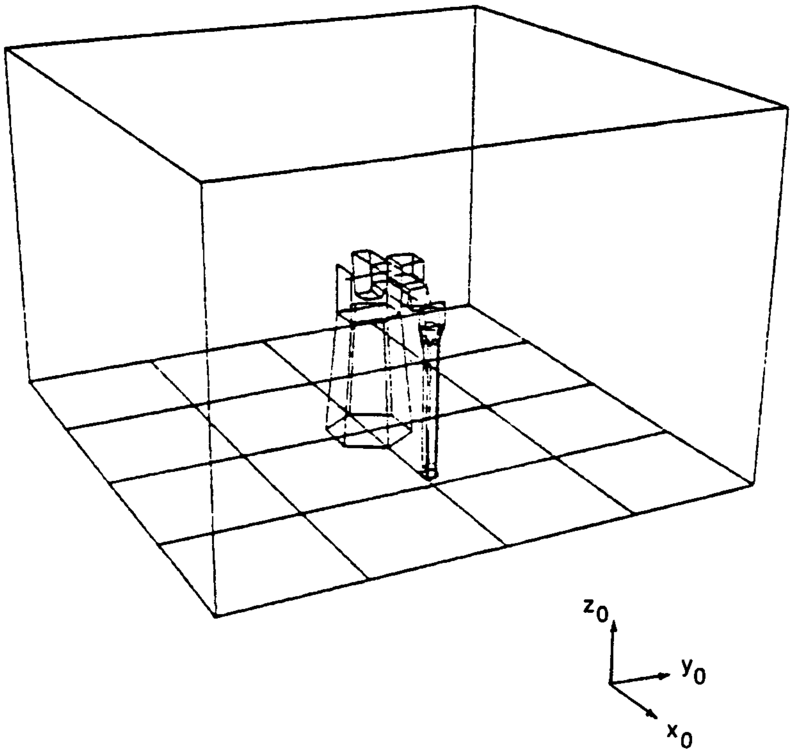
The open-loop control analysis done using FLEXARM shows that the stability margins of the system are greatly reduced by the link flexibility.



Analytically Predicted Open-Loop Transfer Function for Base Control Loop

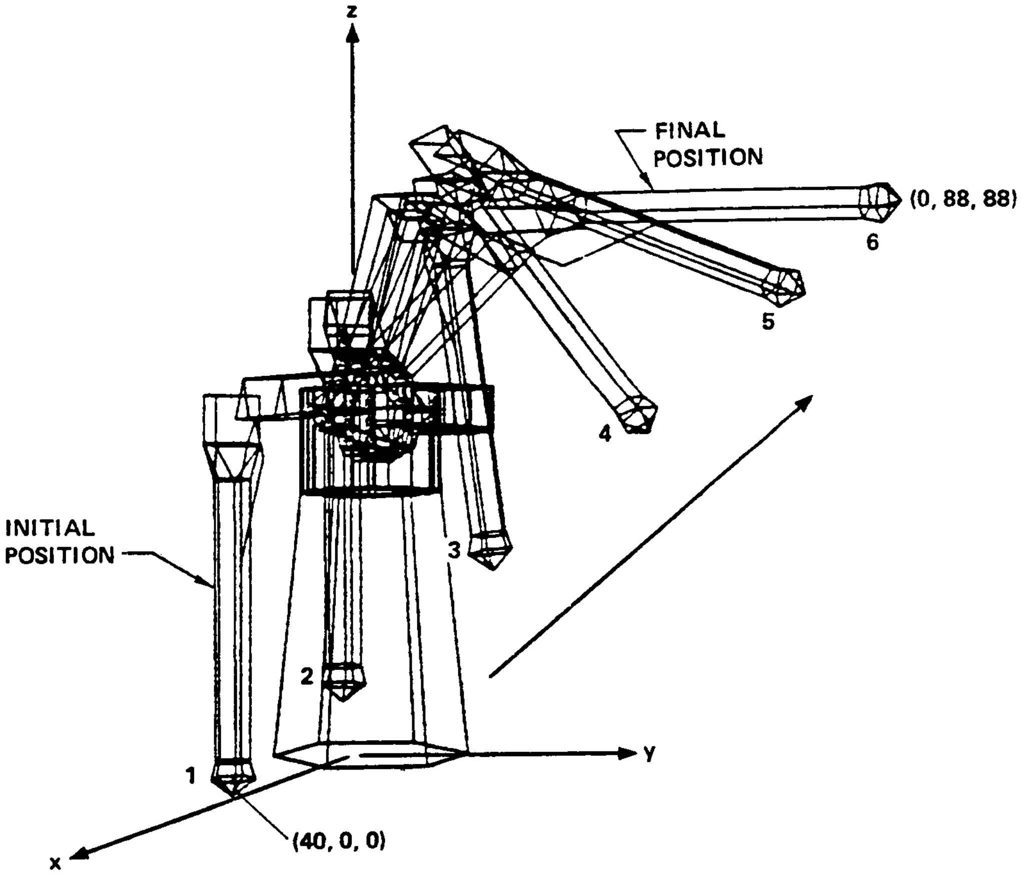
The next group of figures shows the results for FLEXARM simulation of a typical large motion manipulator move.

First, we see the manipulator in its initial position. It will start here from rest. This figure is typical of the computer graphics output mode of FLEXARM. It will then move to its final position with the tip traveling along a straight line in three-dimensional space.

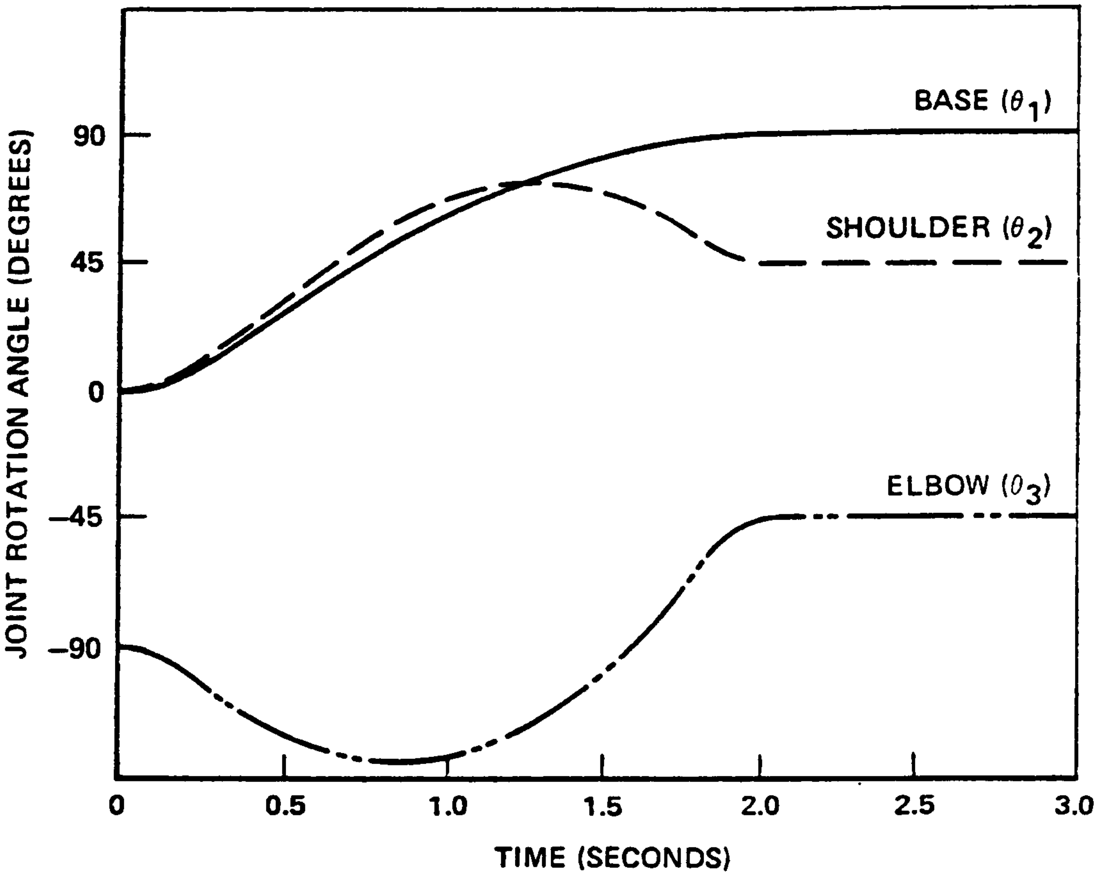


Rotated View of Manipulator in Workspace

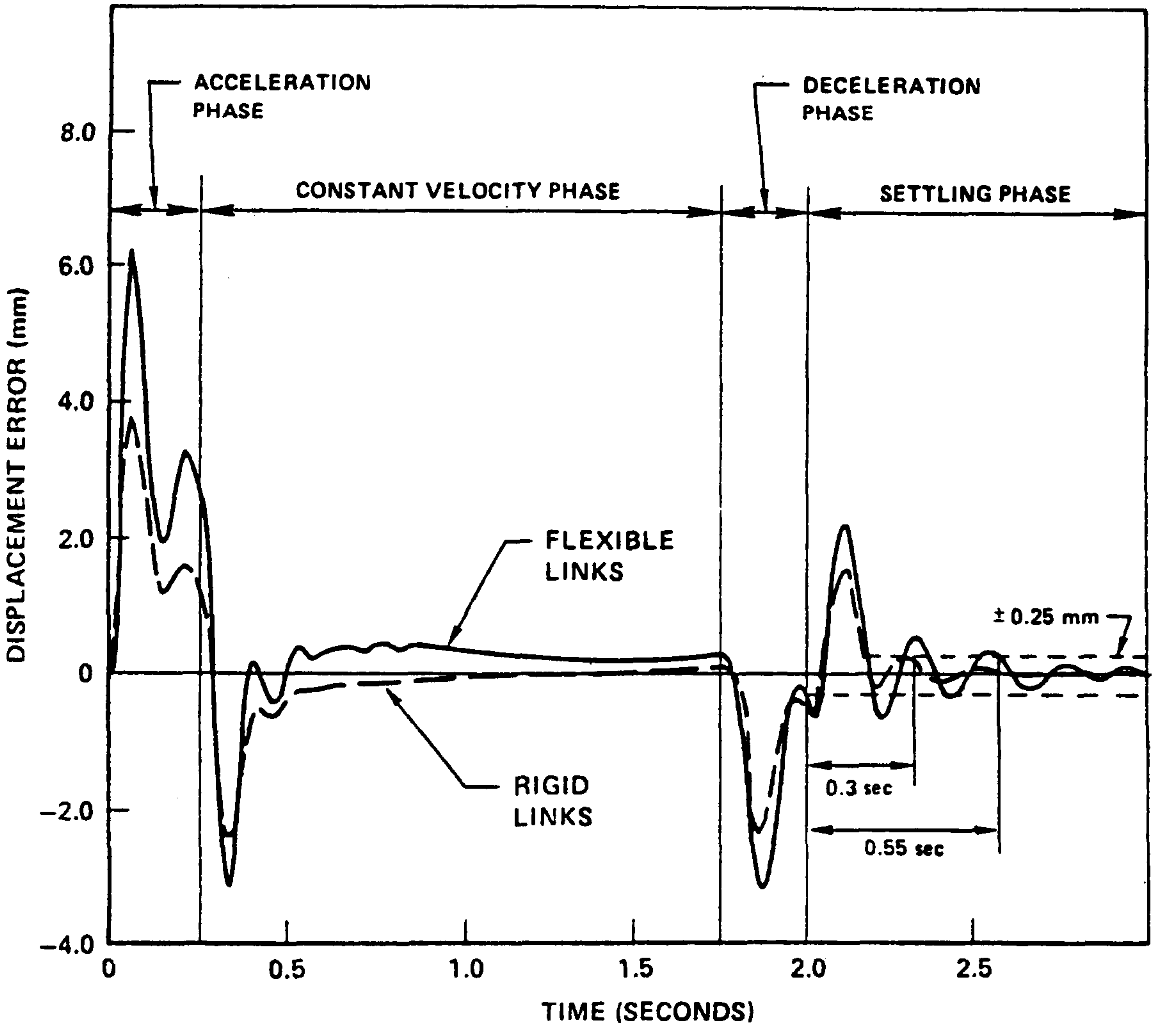
Here the several positions of the manipulator are along its straight line path. As it is standard for many commercial systems, the manipulator tip is commanded to move along its path at a constant acceleration until a constant velocity is reached. It then moves at that constant velocity and then at some point it decelerates to its final position.



As shown here, the joint motions for a simple straight line move are complex functions of time because of the nonlinear kinematic transformations. These joint angles are required as position inputs to the manipulator control systems. The T3R3 is capable of using both velocity and acceleration feed forward signals as well.



The resulting error in the tip position, measured from its nominal position, is shown here as a function of time. Both the response for a "rigid" system and a flexible system are shown. In both cases there are relatively large errors during the acceleration and deceleration phases of the manipulator's motion. For the rigid case most of the error can be attributed to the compressibility of the hydraulic fluid used in the system's drives. The error of the flexible case is significantly larger than that of the rigid case. An important aspect to be noted in this figure is that the time required for the flexible manipulator to settle within its error specification of 0.25 mm at the end of the motion is nearly twice that for the rigid link system. This increased settling time can have a very substantial impact on the productivity of the system in many practical applications. It might also be noted that the results of the studies predict that flexibility of the floor on which the T3R3 is supported can have a very significant effect on the systems performance. In fact, if the floor concrete is less than 4 inches thick, the system can exhibit unstable behavior in certain manipulator positions.



Currently, control algorithms are not available which will effectively control the highly nonlinear dynamic behavior of flexible manipulators. Substantial research on this problem is now being done, but it is a difficult problem.

SOLUTIONS TO FLEXIBILITY PROBLEM

SHORT TERM:

USE OF NEW MATERIALS AND DESIGN CONFIGURATIONS TO MAKE MANIPULATORS LIGHTER - YET MORE RIGID.

LONG TERM:

THE DEVELOPMENT OF CONTROL SYSTEMS TO COMPENSATE MANIPULATOR FLEXIBILITY - AND IDEALLY EXPLOIT IT TO ACHIEVE ULTRA-HIGH SYSTEM PERFORMANCE.

CONCLUSIONS

* FLEXIBILITY CAN PLAY AN IMPORTANT ROLE IN THE DYNAMIC PERFORMANCE OF HIGH-PERFORMANCE MACHINE SYSTEMS.

* EFFICIENT AND ACCURATE FEM METHODS CAN BE DEVELOPED FOR THE MODELING OF NONLINEAR MACHINE SYSTEMS, INCLUDING ROBOTIC MANIPULATORS.

* THE CURRENT MANIPULATORS ARE DESIGNED TO AVOID THE PROBLEMS INTRODUCED BY FLEXIBILITY. HOWEVER, THIS SIGNIFICANTLY LIMITS THE PERFORMANCE OF THESE SYSTEMS.

* NEW CONTROL SYSTEM ALGORITHMS ARE REQUIRED TO PERMIT THE DESIGN OF LIGHTWEIGHT HIGH-PERFORMANCE ROBOTIC SYSTEMS. THESE CONTROL ALGORITHMS NOT ONLY SHOULD COMPENSATE FOR SYSTEM FLEXIBILITY BUT THEY SHOULD ALSO EXPLOIT IT!

BIBLIOGRAPHY

1. "Application of Robotics and Artificial Intelligence to Reduce Risk and Improve Effectiveness: A Study for the United States Army," Published by the National Academy Press, Washington, D.C., 1983.

2. Dubowsky, S., "Dynamics and Control of Robotic Manipulators," *Proceedings of the NATO Advanced Study Institute on Robotics and Artificial Intelligence*, Castelveechio Pascoli, Italy, June 26 - July 3, 1983.

3. Sunada, W., and Dubowsky, S., "The Application of Finite Element Methods to the Dynamic Analysis of Flexible Spatial and Co-Planar Linkage Systems," *J. of Mechanical Design*, ASME Trans., Vol. 103, No. 3, pp. 643-651, July 1981.

4. Sunada, W., and Dubowsky, S., "On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators with Elastic Members," *J. of Mechanical Design*, ASME Trans., Vol. 105, No. 1, pp. 42-51, March 1983.

5. Judd, R.P., and Falkenburg, D.R., "Dynamics of Nonrigid Articulated Robot Linkages," *Proc. of the 1983 American Control Conference*, San Francisco, CA, 1983.

6. Singh, R.P., and Likens, P.W., "Manipulator Interactive Design with Interconnected Flexible Elements," *Proc. of the 1983 American Control Conference*, San Francisco, CA, 1983.

7. Book, W.J., "Analysis of Massless Elastic Chains with Servo Controlled Joints," *J. of Dynamic Systems, Measurement, and Control*, ASME Trans., Vol. 107, pp. 187-192, 1979.

8. Book, W., Maizza-Neto, O., and Whitney, D.O., "Feedback Control of Two Beam, Two Joint System with Distributed Flexibility," *J. of Dynamic Systems, Measurement, and Control*, ASME Trans., Vol. 97, No. 4, pp. 424-431, Dec. 1975.

9. Truckenbrodt, A., "Modeling and Control of Flexible Manipulator Structures," *Proc. of IV CISM-IFToMM Symposium on the Theory and Practice of Robots and Manipulators*, Warsaw, Poland, Sept. 1981.

10. Cannon, R.H., and Schmitz, E., "Initial Experiments on the End-Point Control of a Flexible One-Link Robot," *The International Journal of Robotics Research*, The MIT Press, Vol. 3, No. 3, Fall, 1984.

DYNAMICS OF ARTICULATED STRUCTURES

Edward J. Haug
Center for Computer Aided Design
College of Engineering
The University of Iowa
Iowa City, Iowa

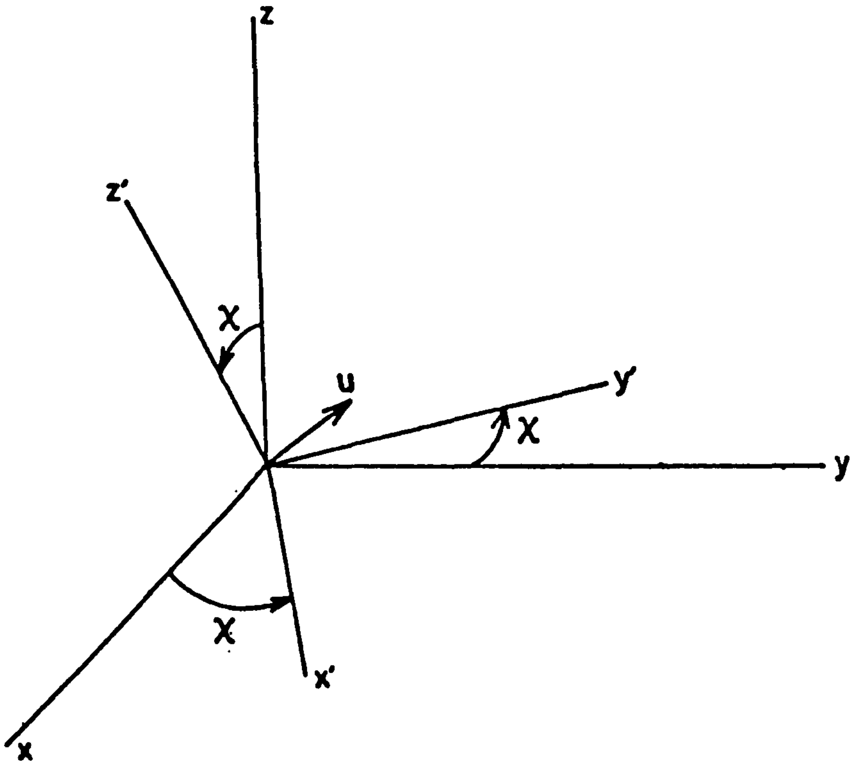
DYNAMICS OF ARTICULATED STRUCTURES

For purposes of this presentation, an articulated structure is defined as an assembly of flexible bodies that may be coupled by kinematic connections and force elements that permit large relative displacement and rotation. Kinematics of such systems is defined using one reference frame for each body in the system and deformation modal coordinates that define displacement fields within flexible bodies. Deformation kinematics are defined by both elastic vibration and static correction deformation modes. Linear elastic deformation is presumed; i.e., a linear stress-strain relation is valid and relative displacements within each elastic component are small enough so that the theory of linear elasticity applies. Coupling of reference and modal coordinates leads to a system of nonlinear equations of motion. Methods of automatically generating and solving these equations of motion are outlined.

- Large Displacement and Rotations of Body Reference Frames (reference coordinates)
- Elastic Vibration and Static Correction Deformation Modes (modal coordinates)
- Coupled Nonlinear Equations in Reference and Modal Coordinates
- Automated Equation Generation and Solution

EULER'S THEOREM

Euler's theorem guarantees existence of a unit vector **u** about which an x'-y'-z' reference frame may be rotated by an angle χ to bring it from a reference x-y-z frame to a general orientation. Components of the unit vector **u** and angle χ of rotation are used to define orientation of a reference frame in space.



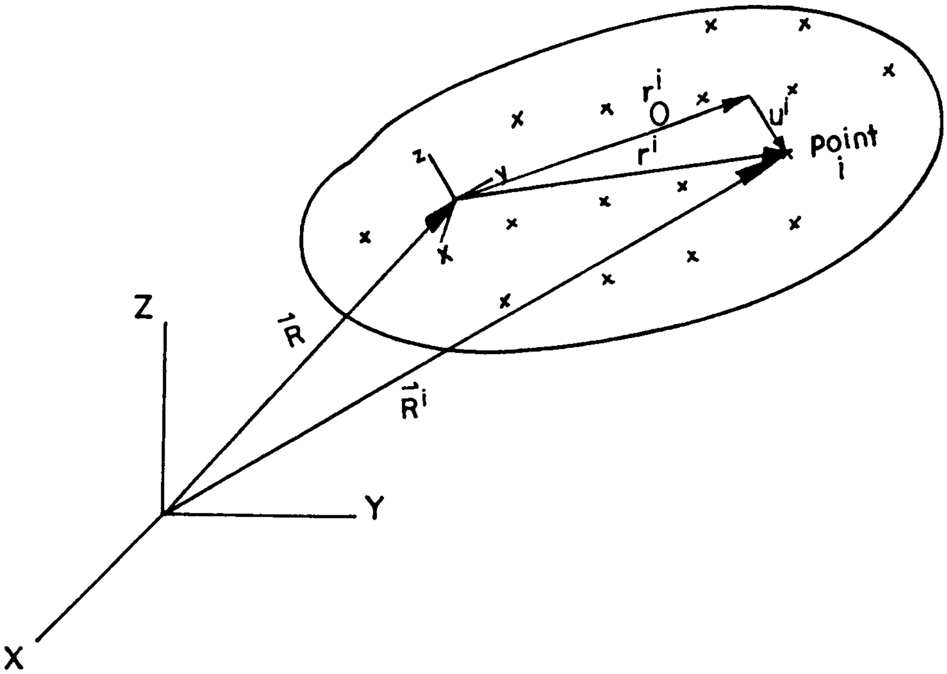
EULER PARAMETERS

A set of four Euler parameters is defined as **p**, as shown on the chart. These four parameters are not independent, since the vector **p** must be a unit vector in R^4. The direction cosine transformation matrix from the x'-y'-z' reference frame to the x-y-z frame is defined as shown. The quadratic nature of terms in the transformation matrix, as functions of Euler parameters, leads to attractive properties when writing velocity and acceleration equations that are needed in the equations of motion. Furthermore, use of Euler parameters avoids singular orientation difficulties that are associated with a set of three rotation parameters, such as Euler angles or Bryant angles [1,2].

- $e_0 = \cos\chi/2, \ \mathbf{e} = \mathbf{u}\sin\chi/2$
- $\mathbf{p} = [e_0, \mathbf{e}^T]^T = [e_0, e_1, e_2, e_3]^T$
- $\mathbf{p}^T\mathbf{p} = \cos^2\chi/2 + \mathbf{u}^T\mathbf{u}\sin^2\chi/2 = 1$
- $\mathbf{s} = \mathbf{A}\mathbf{s}'$, $\mathbf{A} = 2\begin{bmatrix} e_0^2 + e_1^2 - 1/2 & e_1e_2 - e_0e_3 & e_1e_3 + e_0e_2 \\ e_1e_2 + e_0e_3 & e_0^2 + e_2^2 - 1/2 & e_2e_3 - e_0e_1 \\ e_1e_3 - e_0e_2 & e_2e_3 + e_0e_1 & e_0^2 + e_3^2 - 1/2 \end{bmatrix}$

LUMPED MASS FLEXIBLE BODY MODEL

A lumped mass finite-element formulation is used to carry out vibration and static correction mode analysis of each deformable body in an articulated structure. A typical point P^i is defined in the undeformed state of the body by a constant vector $r_0{}^i$. During the process of deformation, this point undergoes displacement u^i in the body reference frame, as shown. Lumped masses m_i at each note in the finite-element model are used in defining kinetic properties of the flexible body [3,4].



KINEMATICS

A vector **u** of modal displacements is defined as a linear combination of a set of deformation modes $\boldsymbol{\phi}^j$, $j = 1,\cdots,m$. The displacement vector $\mathbf{u}^i$ of point i in the body is provided by a projection matrix $\mathbf{P}^i$. As noted earlier, the direction cosine transformation matrix for the reference frame associated with the body is a function of the Euler parameters of that reference frame. Finally, the global position vector $\mathbf{R}^i$ of point i on the body is given as shown.

$u = a_j\phi^j$ - nodal displacement relative to reference frame

$\phi^j = j=1,\cdots,m$ - deformation modes

$u^i = P^i u$ - elastic displacment of mode i

$A(p)$ - direction cosine matrix of reference frame

p - Euler parameters of reference frame

$$R^i = R + A\left(r_0^i + P^i a_j \phi^j\right)$$

VIBRATION AND STATIC CORRECTION MODES

Boundary conditions must be selected for characterizing deformation of flexible components. Since kinematic constraints on bodies in an articulated structure often lead to statically indeterminant sets of boundary conditions, a statically determinant or underdetermined set of boundary conditions is selected for use in vibration analysis. Unit loads associated with deleted kinematic constraints are used to define static correction modes [3]. These calculations are carried out with any standard finite-element code that is capable of generating lumped mass information. Constants that will appear subsequently in the equations of motion are calculated using information generated within the finite-element code.

- Select Boundary Conditions for Flexible Components
- Calculate Natural Vibration Modes
- Calculate Static correction Modes for Deleted Constraints
- Calculate Constants for Equations of Motion

The position relationship derived earlier is differentiated to obtain the global velocity vector of node i in the body, as shown. The time derivative of the direction cosine orientation matrix yields an expression in the time derivative of Euler parameters as shown [2,3]. The velocity vector may thus be written in matrix form for use in derivation of the equations of motion of the system.

$$\dot{R}^i = \dot{R} + \dot{A}r^i + AP^i \phi^j \dot{a}_j$$

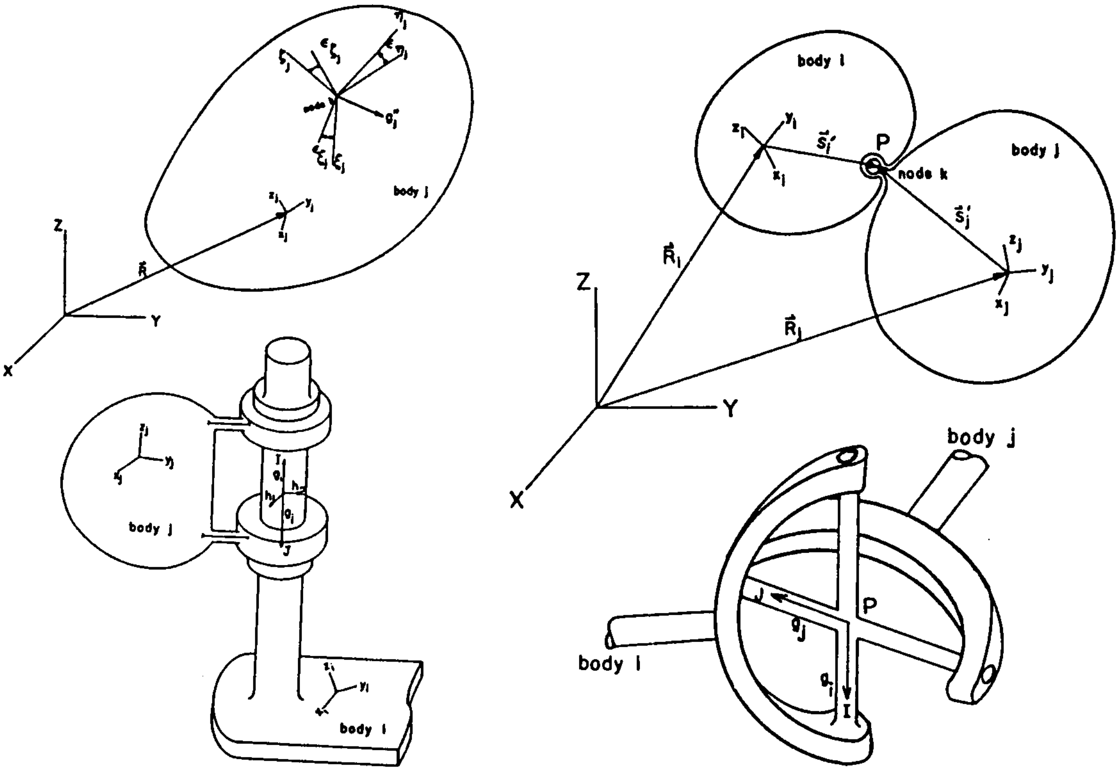
$$\dot{A}r^i = -2E(\underset{\sim}{p})\overset{+}{r}\dot{p}$$

$$\dot{R}^i = \begin{bmatrix} I & -2E\overset{+}{r}^i & A\Psi_t^i \end{bmatrix} \left[\begin{array}{c} \dot{R} \\ \hline \dot{\underset{\sim}{p}} \\ \hline \dot{a} \\ \hline \end{array}\right]$$

$$\Psi_t^i = P^i[\phi^1, \dots, \phi^m]$$

KINEMATIC CONSTRAINTS

A variety of kinematic couplings between flexible bodies is derived in Refs. 2 and 3. Joint definition ξ_j η_j ζ_j reference frames are fixed to the deformable body to define information required to write kinematic constraint equations associated with each joint in the system. Shown on the chart below are spherical, revolute, and universal joints, for which constraint equations may be found in Ref. 3.



KINETICS

The kinetic energy of a flexible body may be written in terms of time derivatives of reference frame generalized coordinates and modal coordinates, as shown. Since vectors r^i and matrices **E**, **G**, and **A** are nonlinear functions of Euler parameters, the mass matrix of the flexible body is a nonlinear function of generalized coordinates, as a result of geometric nonlinearities in the system kinematics.

- Lumped masses m_i, i=1,···, n

$$T = \tfrac{1}{2}\begin{bmatrix} \dot{R} \\ \dot{p} \\ \dot{a} \end{bmatrix}^T M \begin{bmatrix} \dot{R} \\ \dot{p} \\ \dot{a} \end{bmatrix}$$

$$M = \left[\begin{array}{c|c|c} \sum_{i=1}^{N} m_i & -2E \sum_{i=1}^{N} m_i r^{+i} & A \sum_{i=1}^{N} m_i \Psi_t^i \\ \hline & 4G^T\left(\sum_{i=1}^{N} m_i \tilde{r}^{i^T} \tilde{r}^i \right)G & -2 \sum_{i=1}^{N} m_i G^T \tilde{r}^{i^T} \Psi_t^i \\ & & + 2p \sum_{i=1}^{N} m_i r^{i^T} \Psi_t^i \\ \hline \text{symmetric} & & \sum_{i=1}^{N} m_i \Psi_t^{i^T} \Psi_t^i \end{array}\right]$$

FORCES

The strain energy U of the flexible body may be written explicitly in terms of a modal stiffness matrix K_{aa}, as shown. Generalized forces defined through direct application of the definition of virtual work lead to nonlinear algebraic expressions in generalized coordinates, as shown. These forces include both externally applied forces and forces of interaction due to compliant couplings between bodies and feedback control actuators.

$$U = \tfrac{1}{2}\begin{bmatrix} R \\ p \\ a \end{bmatrix}^T \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & K_{aa} \end{array}\right] \begin{bmatrix} R \\ p \\ a \end{bmatrix}$$

$$\delta W = \sum_{i=1}^{N} F^{i^T} \left[I \quad -2A\tilde{r}^i G \quad A\Psi_t^i \right] \begin{bmatrix} \delta R \\ \delta p \\ \delta a \end{bmatrix}$$

$$= \left[\sum_{i=1}^{N} F^{i^T} \quad \sum_{i=1}^{N} -2F^{i^T} A\tilde{r}^i G \quad \sum_{i=1}^{N} F^{i^T} A\Psi_t^i \right] \begin{bmatrix} \delta R \\ \delta p \\ \delta a \end{bmatrix}$$

$$= \left[Q_R^T \quad Q_p^T \quad Q_a^T \right] \begin{bmatrix} \delta R \\ \delta p \\ \delta a \end{bmatrix}$$

EQUATIONS OF MOTION OF A SINGLE BODY

The equations of motion of an unconstrained individual flexible body are shown below [3]. The system of equations for an articulated structure that is made up of multiple bodies connected by kinematic constraints is developed [3] using the Lagrange multiplier form of multi-body system dynamics [1,2]. Evaluation of individual terms appearing in the coefficient matrix of accelerations and on the right side of the equations of motion is derived by expanding the expressions shown and calculating constant coefficients associated with deformation modes and mass distribution.

$$
\left[\begin{array}{c|c|c|c}
\sum\limits_{i=1}^{N} m_i & -2\sum\limits_{i=1}^{N} m_i \overset{+}{r}^i & A\sum\limits_{i=1}^{N} m_i \Psi_t^i & 0 \\
\hline
 & 4G^T\left(\sum\limits_{i=1}^{N} m_i \tilde{r}^{i^T}\tilde{r}^i\right)G & \begin{array}{c} -2G^T\sum\limits_{i=1}^{N} m_i \tilde{r}^{i^T}\Psi_t^i \\ +2p\sum\limits_{i=1}^{N} m_i r^{i^T}\Psi_t^i \end{array} & 2p \\
\hline
 & \text{symmetric} & \sum\limits_{i=1}^{N} m_i \Psi_t^{i^T}\Psi_t^i & 0 \\
\hline
 & & & 0
\end{array}\right]
\left[\begin{array}{c}
\ddot{R} \\ \hline \ddot{p} \\ \hline \ddot{a} \\ \hline \lambda
\end{array}\right]
=
\left[\begin{array}{c}
\sum\limits_{i=1}^{N} F^i \\ \hline
\sum\limits_{i=1}^{N} 2G^T\tilde{r}^{i^T}A^T F^i \\ \hline
\sum\limits_{i=1}^{N} (A\Psi_t^i)^T F^i \\ \hline
0
\end{array}\right]
$$

$$
+\left[\begin{array}{c}
2\dot{E}\left(\sum\limits_{i=1}^{N} m_i \overset{+}{r}^i\right)\dot{p} - 4E\left(\sum\limits_{i=1}^{N} m_i \dot{\overset{+}{r}}^i\right)\dot{p} \\
\hline
-8\dot{G}^T\left(\sum\limits_{i=1}^{N} m_i \tilde{r}^{i^T}\tilde{r}^i\right)G\dot{p} + 4\dot{G}^T\left(\sum\limits_{i=1}^{N} m_i \tilde{r}^{i^T}\Psi_t^i\right)\dot{a} + 2G^T\left(\sum\limits_{i=1}^{N} m_i \dot{\tilde{r}}^{i^T}\Psi_t^i\right)\dot{a} \\
-2p\left(\sum\limits_{i=1}^{N} m_i \dot{r}^{i^T}\Psi_t^i\right)\dot{a} - 4G^T\frac{d}{dt}\left(\sum\limits_{i=1}^{N} m_i \tilde{r}^{i^T}\tilde{r}^i\right)G\dot{p} \\
\hline
-K_{aa}a + 2\left(\sum\limits_{i=1}^{N} m_i \dot{\tilde{r}}^{i^T}\Psi_t^i\right)^T G\dot{p} - 2\left(\sum\limits_{i=1}^{N} m_i r^{i^T}\Psi_t^i\right)^T \dot{p}^T\dot{p} \\
+ 4\left(\sum\limits_{i=1}^{N} m_i \Psi_t^i\right)^T (G\dot{p})^T (G\dot{p}) r^i + 2\left(\sum\limits_{i=1}^{N} m_i \Psi_t^i\right)^T (G\dot{p})^T \Psi_t^i \dot{a} \\
\hline
-2\dot{p}^T\dot{p}
\end{array}\right]
$$

CONSTANTS FROM FINITE-ELEMENT MODEL

Nine sets of constant vectors and matrices shown are calculated, using data generated in the finite-element deformation analysis of each flexible body. These constants are computed using an intermediate processing program [4].

$$C_1 = \sum_{i=1}^{N} m_i r_0^i$$

$$C_2(j) = \sum_{i=1}^{N} m_i \phi_t^{ij} \quad , \qquad j=1,2,\ldots,m$$

$$C_3 = \sum_{i=1}^{N} m_i \tilde{r}_0^i \tilde{r}_0^i$$

$$C_4(j) = \sum_{i=1}^{N} m_i \tilde{r}_0^i \tilde{\phi}_t^{ij} \quad , \qquad j=1,2,\ldots,m$$

$$C_5(j,k) = \sum_{i=1}^{N} m_i \tilde{\phi}_t^{ij} \tilde{\phi}_t^{ik} \quad , \qquad j=1,2,\ldots,m \quad k=1,2,\ldots,m$$

$$C_6(j) = \sum_{i=1}^{N} m_i \tilde{\phi}_t^{ij} r_0^i \quad , \qquad j=1,2,\ldots,m$$

$$C_7(j,k) = \sum_{i=1}^{N} m_i \tilde{\phi}_t^{ij} \phi_t^{ik} \quad , \qquad j=1,2,\ldots,m \quad k=1,2,\ldots,m$$

$$C_8(j) = \sum_{i=1}^{N} m_i r_0^{i^T} \phi_t^{ij} \quad , \qquad j=1,2,\ldots,m$$

$$C_9(j,k) = \sum_{i=1}^{N} m_i \phi_t^{ij^T} \phi_t^{ik} \quad , \qquad j=1,2,\ldots,m \quad k=1,2,\ldots,m$$

NONLINEAR TERMS IN EQUATIONS OF MOTION

Three typical nonlinear terms appearing in the equations of motion presented earlier are shown here, evaluated as linear and quadratic expressions in generalized coordinates. All such terms are coded in a flexible-body module of the Dynamic Analysis and Design System (DADS) computer code. These terms are evaluated at every time step in numerical integration of the coupled system of nonlinear equations of motion.

$$\sum_{i=1}^{N} m_i \overset{+}{r}{}^i = \overset{+}{C}_1 + [\overset{+}{C}_2(j)a_j]$$

$$\sum_{i=1}^{N} m_i \tilde{r}^{i\,T} \tilde{r}^i = -C_3 - \sum_{i=1}^{m} \left[C_4(j) + [C_4(j)]^T\right]a_j - \sum_{j=1}^{m} \sum_{k=1}^{m} C_5(j,k)a_j a_k$$

$$\sum_{i=1}^{N} m_i \tilde{r}^{i\,T} \Psi_t^i = \Big[C_6(1) + \sum_{k=1}^{m} C_7(1,k)a_k,\ C_6(2) + \sum_{k=1}^{m} C_7(2,k)a_k, \ldots,$$

$$C_6(m) + \sum_{k=1}^{m} C_7(m,k)a_k\Big]$$

$$\vdots$$

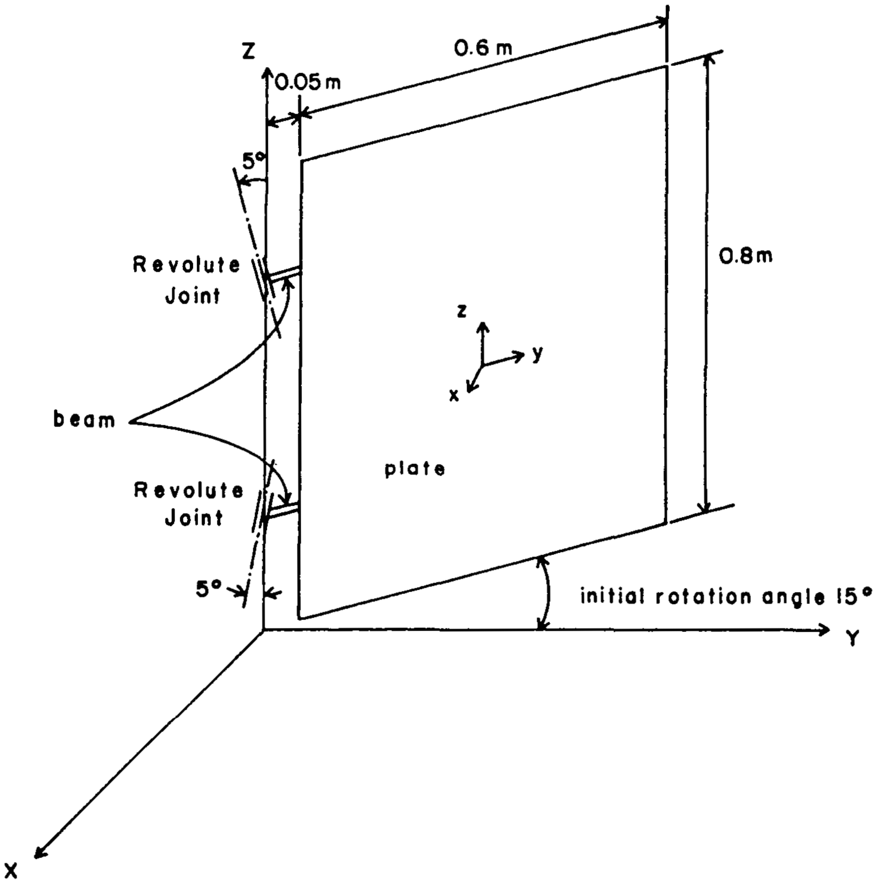
NUMERICAL SOLUTION

A variable order, variable step size numerical integration algorithm is used to compute the solution of differential-algebraic equations of motion for articulated structures. Since step size and order selected by the algorithm reflect the error tolerance required and the frequency of oscillation that develops, integration cost is influenced by selection of deformation modes in the model. Numerical results accumulated to date [4,5] show that prudent selection of a combination of vibration and static correction modes gives reasonable results. Substantial work remains to be done in rational selection of these deformation modes.

- Variable Order, Variable Step Size Numerical Integration
- Integration Cost is a Function of Frequency Content
- Mixed Vibration and Static Correction Modes Give Best Results

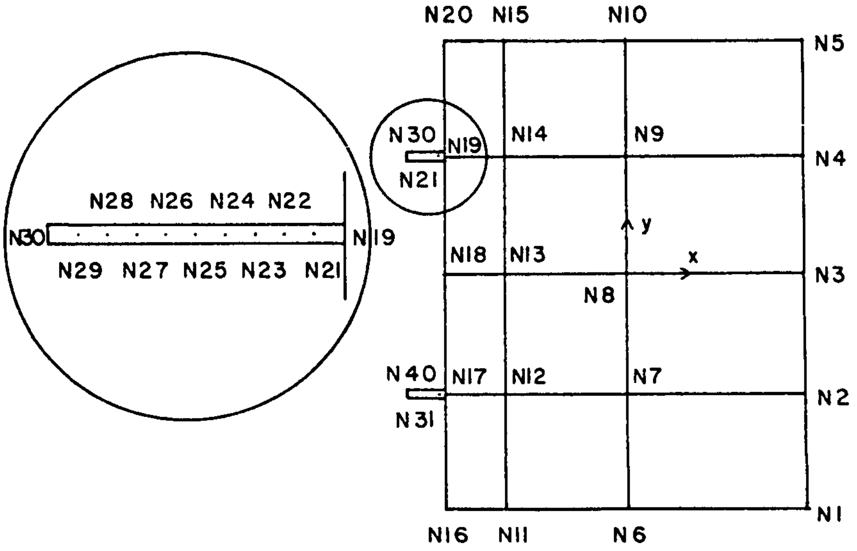
FLEXIBLE HINGE DOOR EXAMPLE

The flexible door structure shown is kinematically coupled to a body that is taken to be rigid ground. The revolute joints shown are misaligned so that there is no deformation when the door structure lies in the Y-Z plane. Any rotation of the door structure leads to deformation of the beam and plate structure of the door, which tends to bring it back to the undeformed state.



FINITE-ELEMENT MODEL OF DOOR STRUCTURE

A modest finite-element model of the door structure, using plate and beam elements, is shown.



N** : Node Number**

Plate ; 12 (Membrane + Bending) Elements

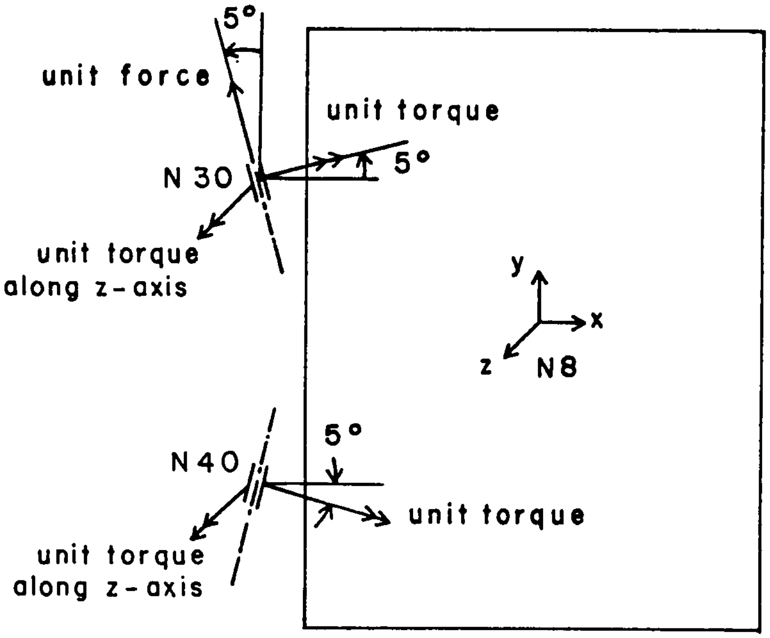
$E = 2.0 \times 10^{12}$ N/m^2

Beam ; 10 Beam Elements at each beam

$E = 2.0 \times 10^{11}$ N/m^2

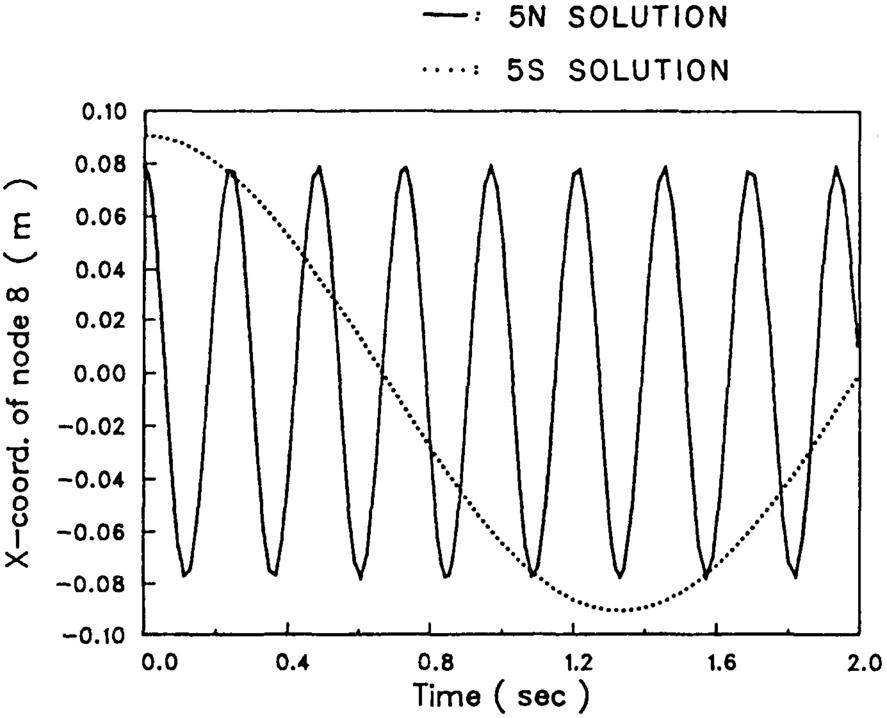
STATIC CORRECTION MODES

Boundary conditions for finite-element analysis are selected so that the center point of the bottom hinge is fixed in space and x- and y-coordinates of the top hinge point are likewise fixed. Five kinematic constraints are thus suppressed, two rotations at the bottom hinge and two rotations and one translation at the top hinge. Unit torques and a unit force are applied to calculate five static correction modes to represent deformation of the structure. Vibration modes are likewise calculated [4].



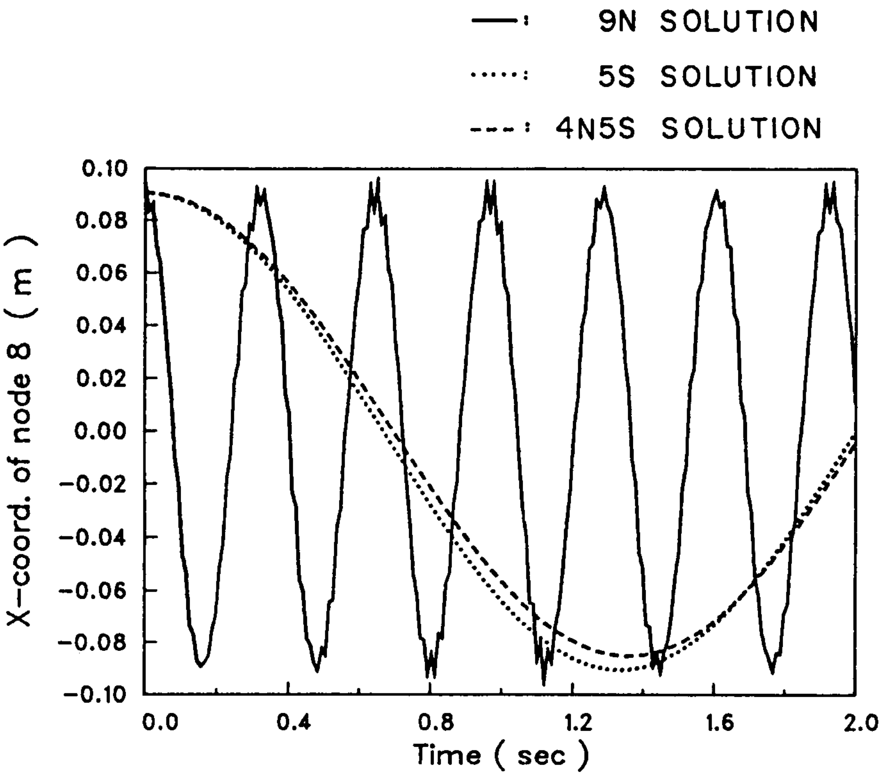
FIRST APPROXIMATE SOLUTION

Two models were used in a preliminary analysis. The first consisted of only five normal vibration modes, with numerical results for the X-coordinate of the center of the door shown as a solid line. A five static correction mode approximate solution is shown with a dotted line, reflecting much lower frequency of vibration of the door structure. To evaluate reasonables of predictions, additional modes and combinations of modes are selected.



SECOND APPROXIMATE SOLUTION

The nine normal mode solution shown reflects a somewhat lower overall vibration frequency, but is still unreasonable. As shown by the dotted and dashed curves, adding four normal vibration modes to the five static correction modes yields only a slight change in the prediction obtained from purely five static correction modes. This suggests that static correction modes dominate the dynamics of this example.



COMPARISON OF COMPUTER SIMULATION TIMES

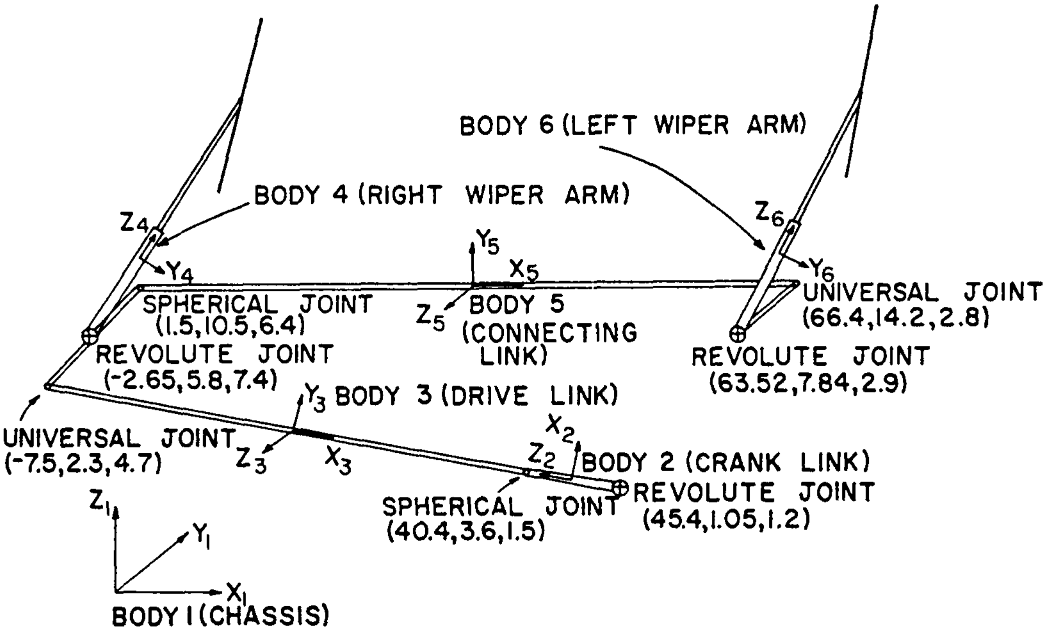
As shown in the table below, retention of a substantial number of high-frequency normal vibration modes leads to very small step size and ultimately exceptionally large computer times. The computer times indicated are on a heavily loaded Prime 750 supermini computer.

Comparison of Simulation Times

Type	T end [sec]	CPU [sec]	RMS integration stepsize [sec]
5S solution	2.0	106	0.47812E-01
5N solution	2.0	401	0.90934E-02
9N solution	2.0	7471	0.65035E-03
4N5S solution	2.0	7281	0.75491E-03

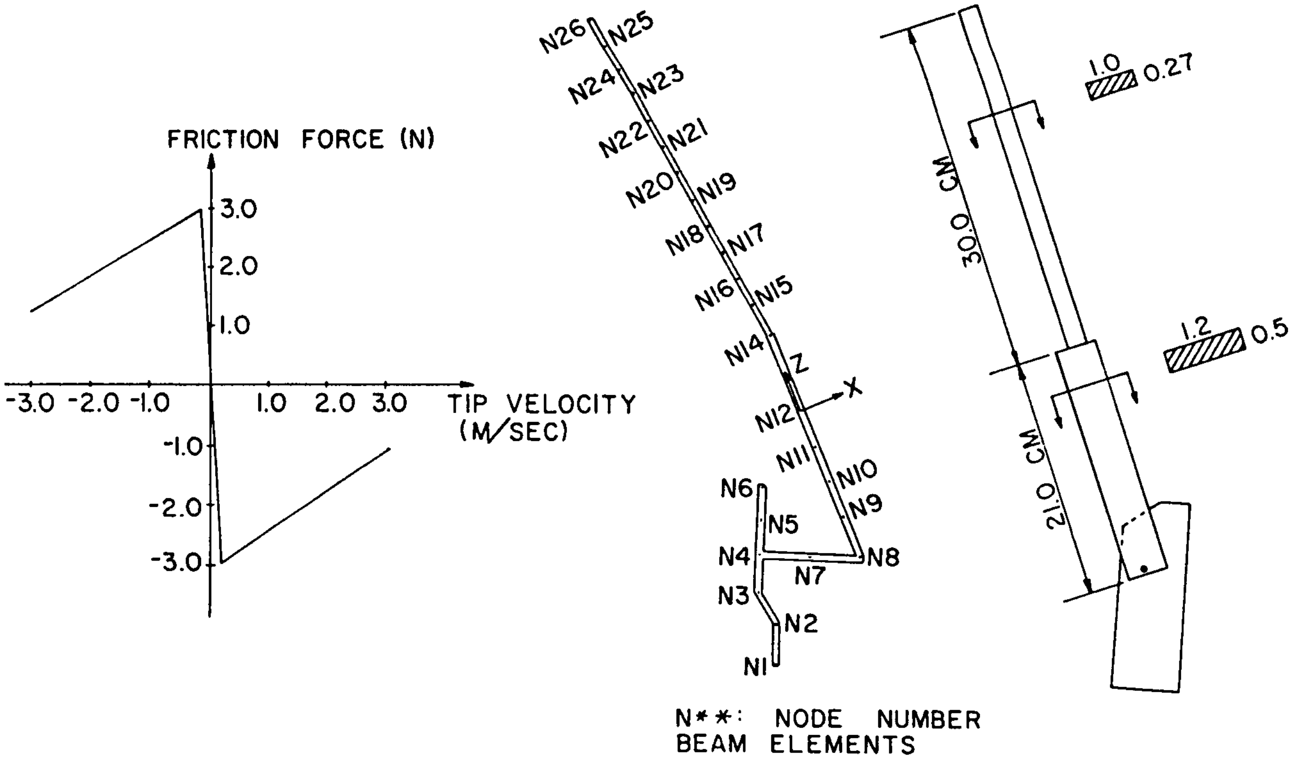
WINDSHIELD WIPER APPLICATION

The schematic shown is a model of an automotive windshield wiper assembly, in which the crank-link and two connecting links are taken as rigid. The left and right wiper arms are modeled as flexible bodies. The mechanism is driven by applying a torque to the crank link that is a function of motor speed.



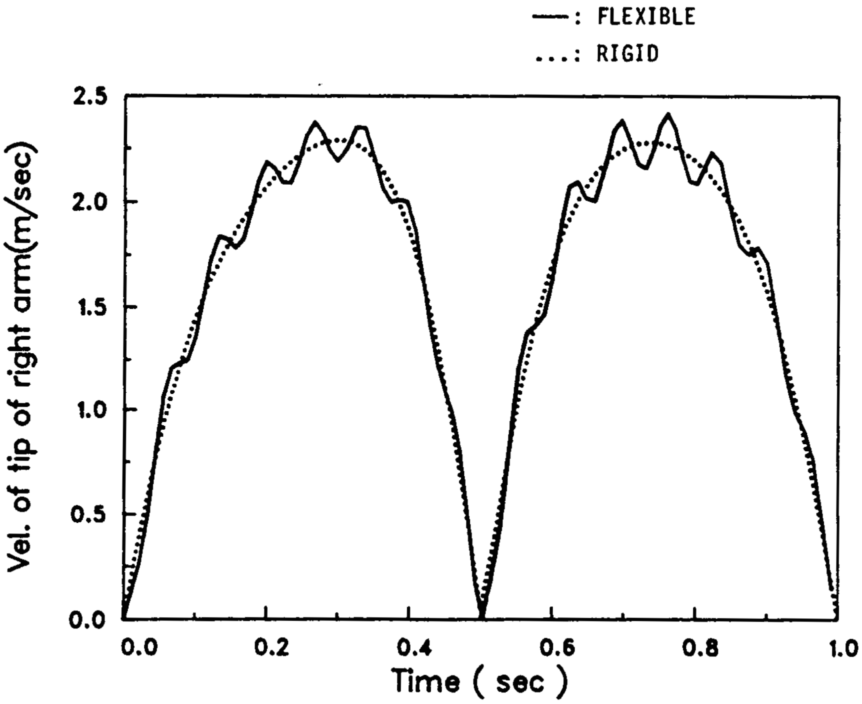
FINITE-ELEMENT MODEL OF WIPER ARMS

A modest Beam finite-element model of each wiper arm is constructed as shown. Friction torque, as a function of wiper tip velocity, is introduced as a force acting in the system, as shown.



NUMERICAL RESULTS FOR WIPER ARM TIP VELOCITY

A flexible body solution shown in the solid line predicts vibration at a frequency of approximately fourteen cycles per second, relative to essentially the same gross motion predicted by a rigid body model of the windshield wiper mechanism. Experimental results with the actual system indicate an oscillation frequency of approximately thirteen cycles per second, very close to that predicted by the articulated structure model.



STATUS AND DEVELOPMENTS

The DADS flexible system dynamics code is now functioning and has been used to analyze a number of small and intermediate scale applications. A commercial version of the software is expected to be available from Computer Aided Design Software Incorporated, of Oakdale, Iowa. Extensions are currently under way to enhance capability of the code to represent selected aspects of space structure dynamics.

- DADS Flexible Code Is Now Functioning
- A Commercial Version of The Code Will Be Available Late In 1985
- Extensions Are Under way To Enhance Capabilities For Space Structure Dynamics

REFERENCES

1. Wittenburg, J., Dynamics of Systems of Rigid Bodies, B. G. Teuber, Stuttguard, 1977.
2. Haug, E. J., Computer Aided Kinematics and Dynamics of Mechanical Systems, Allen and Bacon Publishers, 1989.
3. Yoo, W. S. and Haug, E J., "Dynamics of Articulated Structures, Part I: Theory," Journal of Structural Mechanics, vol. 14, no. 1, 1986, pp. 105-126.
4. Yoo, W. S. and Haug, E. J., "Dynamics of Articulated Structures, Part II: Computer Implementation and Applications," Journal of Structural Mechanics, vol. 14, no. 2, 1986, pp. 177-189.
5. Yoo, W. S. and Haug, E. J., "Dynamics of Flexible Mechanical Systems Using Vibration and Static Correction Modes," Journal of Mechanisms, Transmissions, and Automation in Design, vol. 108, no. 3, September 1986, pp. 315-322.

N89-24665

MODAL REDUCTION STRATEGIES FOR INTERCONNECTED FLEXIBLE BODIES SIMULATION

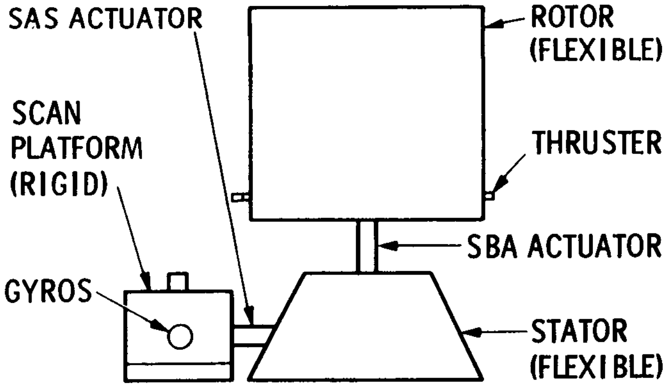
F. O. Eke and G. K. Man
Jet Propulsion Laboratory
Pasadena, CA

INTRODUCTION

MULTI-BODY DYNAMICS PROGRAMS REQUIRE CHARACTERIZATION OF EACH BODY

- RIGID BODY: GEOMETRY AND MASS PROPERTIES
- FLEXIBLE BDOY
  - EXACT TYPE OF INPUT DEPENDS ON PROGRAM
  - ALL INVOLVE MODAL CHARACTERISTICS IN SOME FORM
  - ALWAYS NEED FOR MODAL TRUNCATION
  - SYSTEMATIZE TRUNCATION PROCEDURE

GALILEO SPACECRAFT

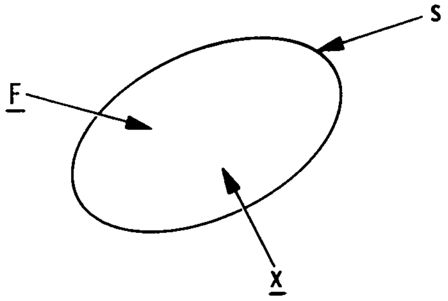


- ACTUATORS: SBA, SAS, THRUSTERS
- SENSORS: GYROS, CLOCK AND CONE ENCODERS, SUN SENSOR, STAR SCANNER
- CLOCK (SBA) CONTROL LOOP IS ACTIVE DURING ALL ATTITUDE CONTROL MANEUVERS
  - CLOCK CONTROLLER BANDWITCH$\cong$0.5 Hz
  - GYRO ROLLOFF FREQUENCY$\cong$ 15 Hz
- NEED "ADEQUATE" MODEL OF PLANT FOR DESIGN AND SIMULATION

TRUNCATION CRITERIA

- CONTROL SYSTEM SPECIFICATIONS CAN SET TRUNCATION CRITERIA AT SYSTEM LEVEL ONLY
  - SYSTEM MODE WITH FREQUENCY ABOVE 15Hz CAN BE DROPPED
  - ELIMINATE MODES THAT DO NOT INTERACT "STRONGLY" WITH THE CONTROL SYSTEM

SYSTEM LEVEL TRUNCATION



METHOD

$$M\ddot{x} + Kx = F \quad (1)$$

$$x = \phi z \quad (2)$$

$$X(s) = \underbrace{\phi(s^2 I + \omega^2)^{-1} \; \phi^T}_{D} F(s) \quad (3)$$

FOR RESPONSE AT i LOCATION DUE TO STEP INPUT AT j LOCATION,

$$X_i(s) = DijFj(s) = \sum_{k=1}^{m} \left\{ \phi_{ik} \; \phi_{jk} \; A/ \left[s\,(s^2 + \omega_k^{\,2}) \right] \right\} \quad (4)$$

SYSTEM LEVEL TRUNCATION (CONT'D)

CONTRIBUTION OF kth MODE TO RESPONSE:

$$X_i^k(s) = \phi_{ik} \; \phi_{jk} \, A/\left[s(s^2+\omega_k^2)\right] \qquad (5)$$

OR

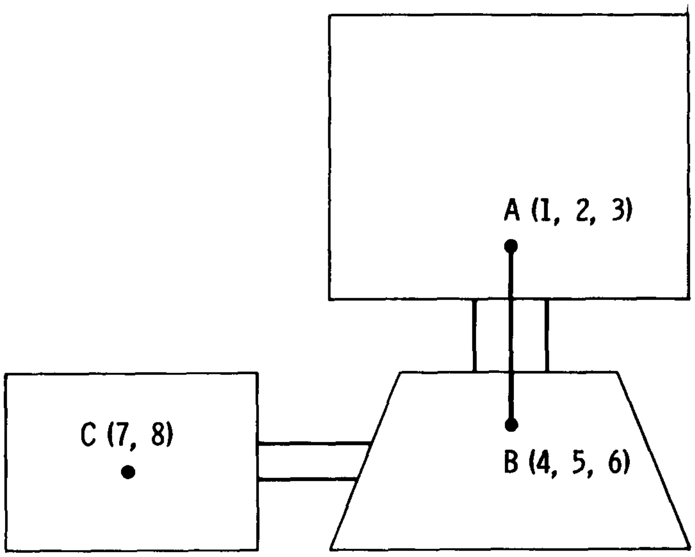
$$X \, x_i^k\,(t) = (\,\phi_{ij} \; \phi_{jk} \, A/\omega_k^2) \;\; \left[1 - \mathrm{COS}\,(\,\omega_k t)\right] \qquad (6)$$

SINUSOIDAL RESPONSE WITH PEAK-TO-PEAK AMPLITUDE TO

$$X_i^k = 2 \; \phi_{ik} \;\; \phi_{jk} \, A/\omega_k^2 \qquad (7)$$

A MEASURE OF IMPORTANCE OF MODE K

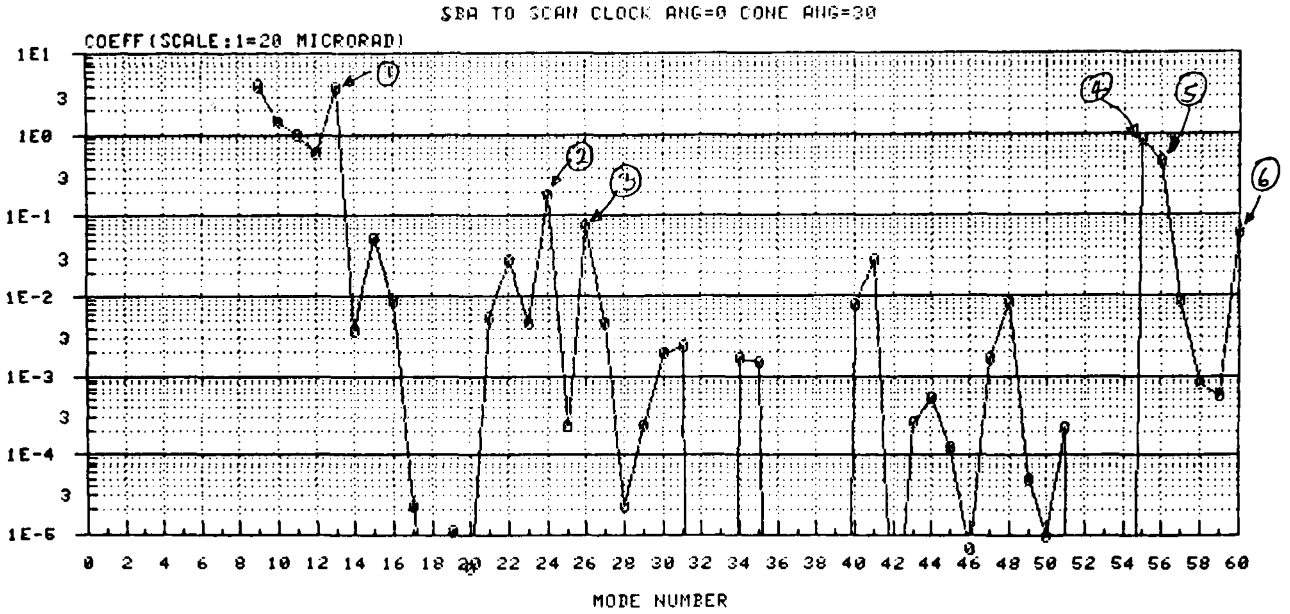
APPLICATION TO GALILEO



APPLICATION TO GALILEO (CONT'D)

- AVAILABLE DATA
  - EIGENVALUES, EIGENVECTORS FOR UP TO 60 MODES
- PLOT MODAL INFLUENCE COEFFICIENTS
- DISCARD MODES WITH "LOW" COEFFICIENTS
- USE BODE PLOT TO CHECK RESULTS

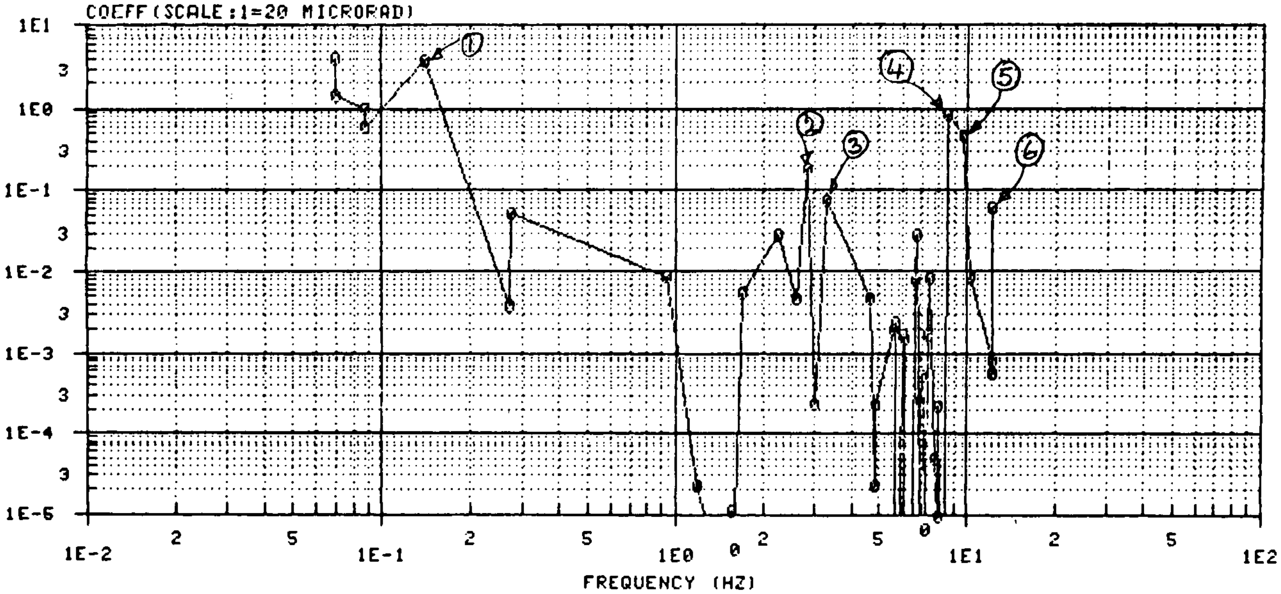
MODAL INFLUENCE COEFFICIENTS



ORIGINAL PAGE IS OF POOR QUALITY

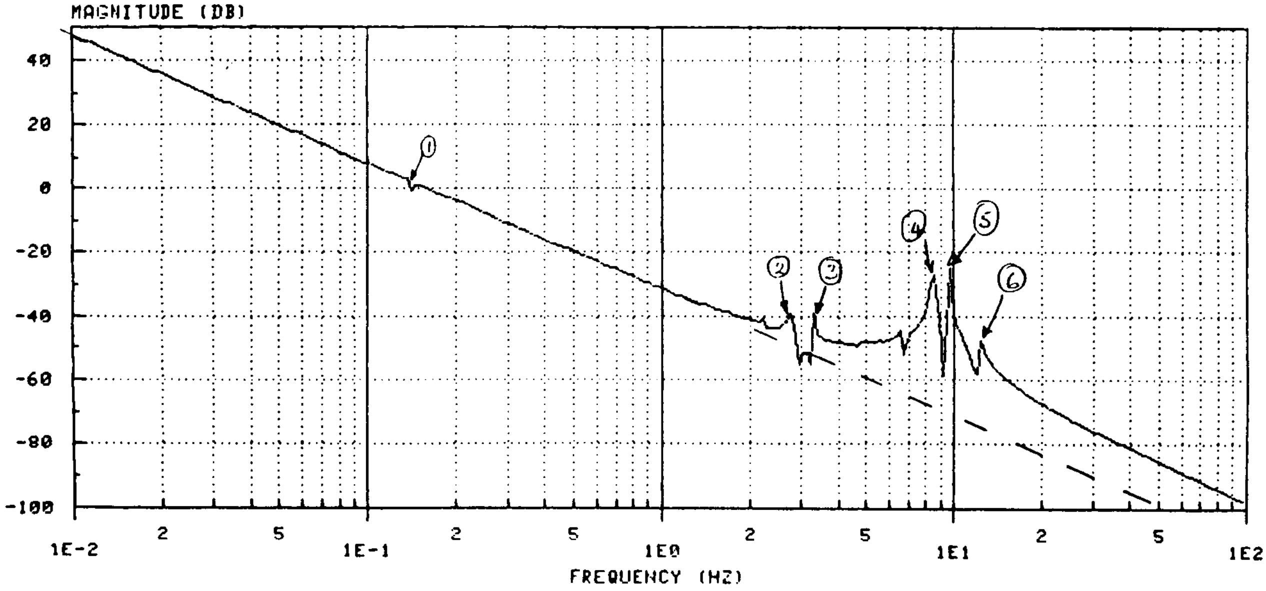
MODAL INFLUENCE COEFFICIENTS

SBA TO SCAN CLOCK ANG=0 CONE ANG=30

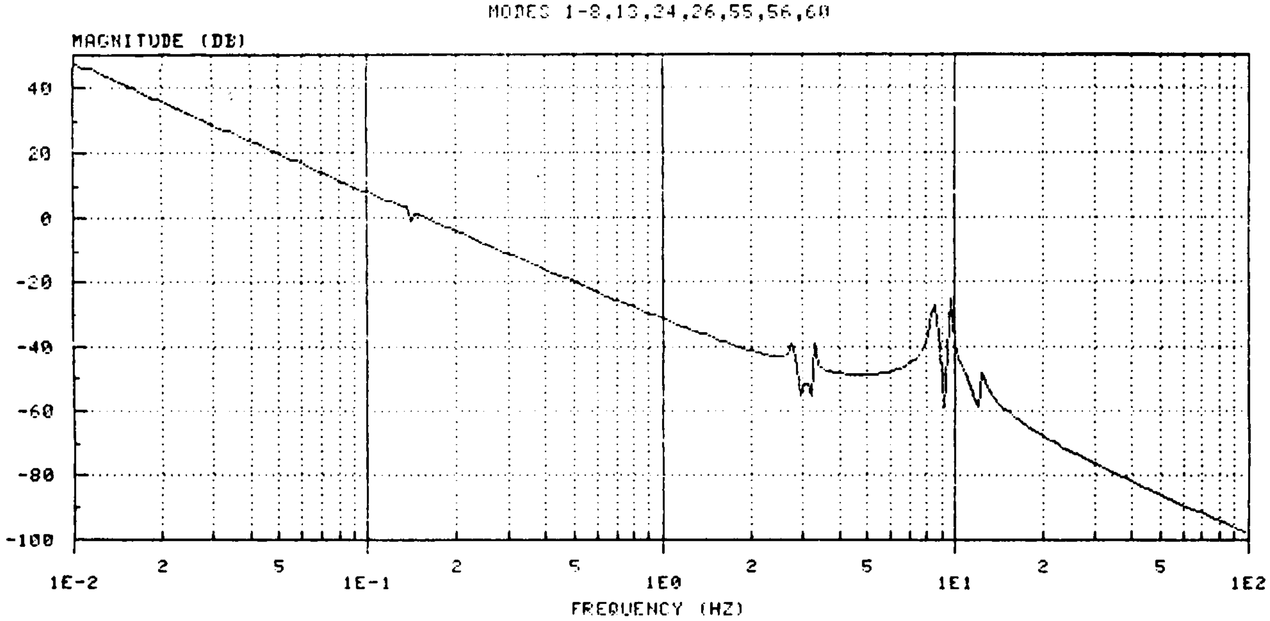


BODE PLOT OF PLANT ALPHA = 0 BETA = 30

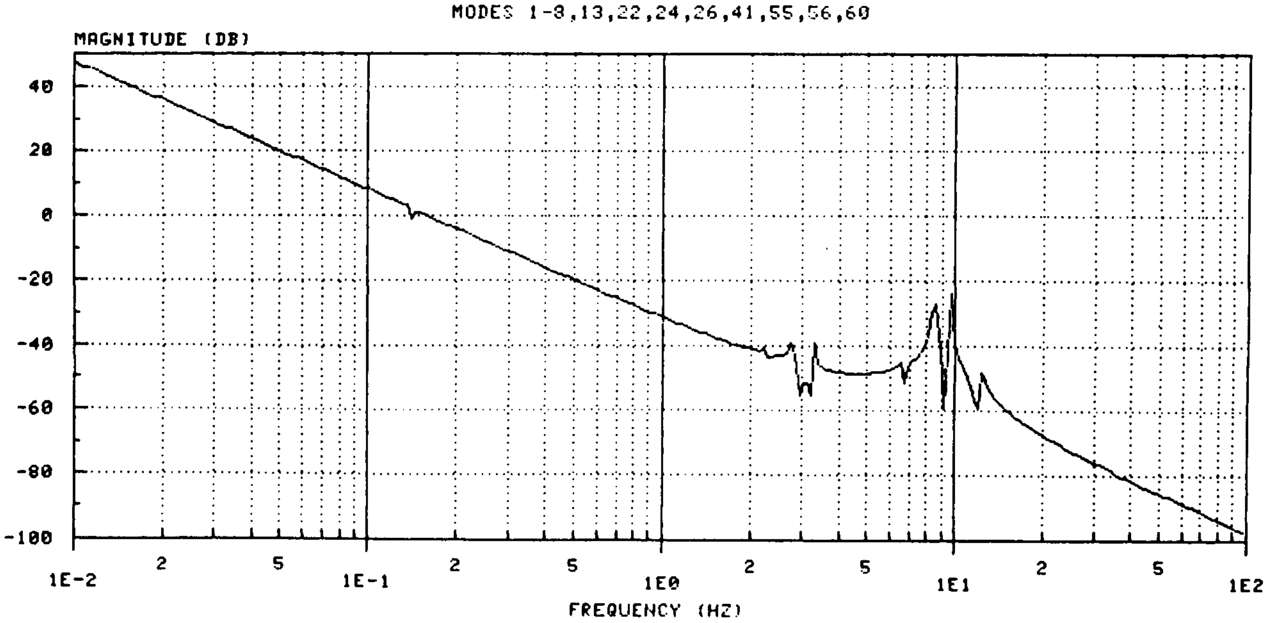
MODES 1-60 .25% DAMPING



BODE PLOT OF PLANT CLOCK = 0 CONE = 30



BODE PLOT OF PLANT ALPHA = 0 BETA = 30

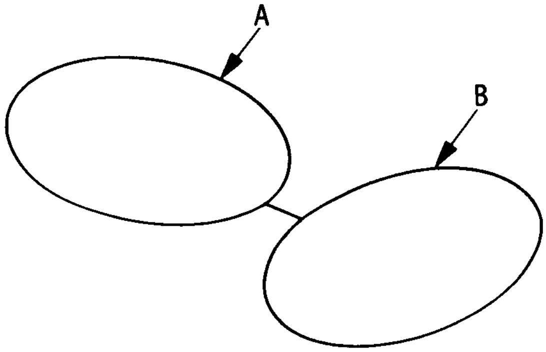


TRUNCATION AT COMPONENT LEVEL

- AVAILABLE
  - COMPONENT "FREE-FREE" MODES
  - SYSTEM MODES TO BE RETAINED
- PROBLEM
  - DETERMINATION OF "IMPORTANT" COMPONENT FREE-FREE MODES FROM KNOWLEDGE OF SYSTEM MODES
- SOLUTION
  - RETAIN THOSE COMPONENT MODES THAT "CONTRIBUTE SUBSTANTIALLY" TO IMPORTANT SYSTEM MODES

COMPONENT LEVEL TRUNCATION (CONT'D)

$$\left.\begin{aligned} M_A \ddot{x}_A + K_A x_A &= F_A \\ x_A &= \phi_A \; q_A \\ I\ddot{q}_A + \omega^2 q_A &= \phi_A^T \; F_A \end{aligned}\right\} \begin{array}{l} \text{BODY A} \\ \text{D.O.F.} = n_A \end{array} \quad (8)$$



$$\left.\begin{aligned} M_B \; \ddot{x}_B + K_B x_B &= F_B \\ x_B &= \phi_B q_B \\ I\ddot{q}_B + \omega_B q_B^2 &= \phi_B^T \; F_B \end{aligned}\right\} \begin{array}{l} \text{BODY B} \\ \text{D.O.F.} = n_B \end{array} \quad (9)$$

$$\left.\begin{aligned} M\ddot{x} + Kx &= F \\ x &= \phi q \\ I\ddot{q} + \omega^2 q &= \phi^T F \end{aligned}\right\} \begin{array}{l} \text{COMBINED} \\ \text{SYSTEM} \\ \text{D.O.F.} = n \leqslant (n_A + n_B) \end{array} \quad (10)$$

COMPONENT LEVEL TRUNCATION (CONT'D)

- SYSTEM AUGMENTED ϕ MATRIX = $\bar{\phi}$
  - SYSTEM MATRIX WITH SOME ROWS REPEATED

$$\left\{\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_{n_A} \\ \hline x_{n_A} \\ \vdots \\ x_{n_A+n_B} \end{array}\right\} = \left[\begin{array}{cccc} \phi_{11} & \phi_{12} & \cdots\cdots & \phi_{1m} \\ \phi_{21} & \phi_{22} & \cdots\cdots & \phi_{2m} \\ & & & \\ \phi_{n_A1} & \phi_{n_A2} & \cdots\cdots & \phi_{n_Am} \\ \hline \phi_{n_{A+1},1} & & \cdots & \phi_{n_{A+1},m} \\ & & & \\ \phi_{n'_1} & \phi_{n'_2} & \cdots\cdots & \phi_{n'm} \end{array}\right] \left\{\begin{array}{c} q_1 \\ q_2 \\ \\ q_{n_A} \\ \hline q_{n_{A+1}} \\ \\ q_{n'} \end{array}\right\} \qquad (11)$$

$\bar{\phi}_A$

$\bar{\phi}_B$

- PARTITION $\bar{\phi}$ INTO $\bar{\phi}_A$ AND $\bar{\phi}_B$

COMPONENT LEVEL TRUNCATION (CONT'D)

- DELETE COLUMNS OF $\bar{\phi}$ THAT CORRESPOND TO SYSTEM MODES THAT WERE DROPPED
- REDUCED ϕ MATRICES: $\hat{\phi}_A$ AND $\hat{\phi}_B$
- USE $\hat{\phi}_A$ AND $\hat{\phi}_B$ AS TRANSFORMATION MATRICES FOR BODIES A AND B RESPECTIVELY
  - $\hat{\phi}_A^T \; M_A \; \hat{\phi}_A \ddot{\hat{q}}_A + \hat{\phi}_A^T \; K_A \; \hat{\phi}\hat{q}_A = \hat{\phi}_A^T \; F_A$ (12)

OR
  - $\hat{M}_A \ddot{\hat{q}}_A + \hat{K}_A \hat{q}_A = \hat{\phi}_A^T F_A$ (13)
  - $\hat{M}_B \ddot{\hat{q}}_B + \hat{K}_B \hat{q}_B = \hat{\phi}_B^T F_B$ (14)

COMPONENT LEVEL TRUNCATION (CONT'D)

- $\hat{M}_A, \hat{K}_A, \hat{M}_B, \hat{K}_B$ NOT NECESSARILY DIAGONAL
- DIAGONALIZE VIA ANOTHER MODAL ANALYSIS
  - $\hat{q}_A = \psi_A \bar{q}_A$ (15)
  - $\hat{q}_B = \psi_B \hat{q}_B$ (16)

→ $I\ddot{\bar{q}}_A + \bar{\omega}_A^2 \bar{q}_A = \psi_A^T \hat{\phi}_A^T F$ (17)

$I\ddot{\bar{q}}_B + \bar{\omega}_B^2 \bar{q}_B = \psi_B^T \hat{\phi}_B^T F$ (18)

COMPONENT LEVEL TRUNCATION (CONT'D)

- $\bar{\omega}_A, \bar{\omega}_B$ ARE DIAGONAL; THEY ARE ALSO SUB-MATRICES OF ω_A, ω_B RESPECTIVELY, AND CONTAIN FREQUENCIES OF COMPONENT MODES TO BE RETAINED

- SIMILARLY $\Phi_A = \hat{\phi}_A \Psi_A$ AND $\Phi_B = \hat{\phi}_B \Psi_B$ ARE SUBMATRICES OF ϕ_A AND ϕ_B, AND CONTAIN THE EIGENVECTORS OF COMPONENT MODES TO BE RETAINED

SUMMARY AND CONCLUSION

- DETERMINE SYSTEM MODES TO BE RETAINED USING
  - AVAILABLE CRITERIA
  - MODAL INFLUENCE COEFFICIENTS
  - BODE
- DESCEND TO COMPONENT LEVEL VIA A TWO-PHASE DIAGONALIZATION PROCESS STARTING WITH SUBMATRICES OF TRUNCATED AUGMENTED SYSTEM MODAL MATRIX

FUTURE WORK

- STREAMLINE SIMULATION CODES – ESPECIALLY DYNAMICS FORMULATION METHOD
- DEVELOP VERY EFFICIENT AND EASILY IMPLEMENTABLE MODEL REDUCTION STRATEGY

N89-24666

COMPUTATIONAL ASPECTS OF MULTIBODY DYNAMICS

K. C. Park
Lockheed Palo Alto Research Laboratory
Palo Alto, California

ABSTRACT

This paper addresses computational aspects impacting the requirements for developing a next-generation software system for flexible multibody dynamics simulation which include: criteria for selecting candidate formulations, pairing of formulations with appropriate solution procedures, need for concurrent algorithms to utilize computer hardware advances, and provisions for allowing open-ended yet modular analysis modules.

Computer Implementation Tasks for Multibody Dynamics Simulator

A successful next-generation multibody dynamics simulator requires a careful evaluation of existing formulations and computational procedures from which pairing of several candidate formulations/solution algorithms should evolve and, if necessary, need for new and/or improved formulations and solution algorithms must be identified. Concurrent with selecting formulations and solution algorithms, considerations must be given to software environment under which the next-generation simulator will be implemented. In addition, the associated hardware systems and their future trend must be incorporated from the outset of the computer implementation planning stage. These aspects are summarized.

Formulations
Solution Procedures
Software Environment
Hardware Systems

Review of Available Formulations

Formulations According to:

- Bodley/Frisch
- Fraeijs de Veubeke
- Hooker/Margoulis/Ho
- Kane/Likins
- Roberson/Wittenburg
- Russel/Jerkovsky

Criteria for Selecting Candidate Formulation

- Efficiency of the Resulting Software Rather Than Simplicity of the Resulting Equations of Motion
- Let Implementation Algorithm Select the Generalized Coordinates Rather Than Case-by-Case User Selection of Them

Review of Available Solution Algorithms

Stiff Differential Integrator(Hindmarsh/Gear)
Differential/Algebraic Solver(Petzold/Lötstedt)
Partitioned Procedures(Park/Felippa)
Semi-Implicit Runge-Kutta methods(Chipman/März)
Impact-Contact Algorithms

Criteria for Selecting Candidate Algorithms

Reliability First, Then Efficiency
Rather Than
Efficiency First, Then Reliability

Minimum User Decision

Current Software and Hardware Environment

Most of the currently available computer programs for simulating multibody dynamics do not have any data base management. As such, the task of data handling remains time consuming and inflexible. In particular, an addition of enhanced capability can present varying difficulties. However, improved computational efficiency has been brought about by vectorization of part of the programs that require intensive computations to generate the discrete dynamical equations and then installing the resulting programs in CRAY-like supercomputers.

An Example of Challenging Deployment Task: 100-Meter Parabolic Truss

Deployed Diameter	=	100 m
Stowed Diameter	=	1.36 m
Core Strut Length	=	4.8 m
Deployed Truss Depth	=	4.0 m
Stowed Package Length	=	4.79 m
Slenderness Ratio of Struts	=	1069

Number of Nodes	=	760
Number of Struts	=	3234

Number of Control Links	=	6468
Number of Slider Joints	=	760
Number of Revolute Joints	=	21,549

Desired Formulations for Next-Generation Simulator

1. System Topology Must Be Presented to the Computer by a General Graphic Theory with Efficient Search Algorithms.

2. Kinematic and Equilibrium Equations of Individual Elements Must Be Generated by Efficient Symbolic Manipulations.

3. Necessary Transformation Matrices for Assembling the System Equations Must be Flexible Enough And Yet Arranged in a Form That Requires a Minimum User Decision and Resulting Always in Nonsingular System.

4. Formulations Should Allow Assembly of System Equations Either With or Without Constraints as Primary Variables.

5. Most Important of All, Modeling of Element Flexibility Should Allow Either Generalized Coordinates or Finite-Element Physical Coordinates.

Outstanding Algorithmic Difficulties

1. Solution Matrix for Fully Implicit Algorithm Becomes Nonsymmetric.

2. Elimination of Constraint Forces Complicates Matrix Profiles. On the Other Hand, Preservation of Constraint Forces as Independent Variables Increases Equation Size.

3. Augmentation of Constraint Equations Introduces Algebraic Equations Which Can Lead to Numerical Drifting in the Solution: Stabilization Becomes Important.

4. Member Flexibility and Joint Friction Introduces High-Frequency Solution Components and Sometimes Severe Nonlinearities.

5. Systematic Selection of Independent Set of Generalized Coordinates Present Formidable Challenge.

6. Determination of Initial Conditions from a Known Partial Set of Initial Conditions Is Often a Difficult Task.

7. Finally, Matching a Particular Formulation with a Most Suitable Solution Algorithm Requires an In-depth Investigation of the Combined Characteristics of the System Equations and Numerical Algorithms.

Recommended Formulation

1. Dual Formulations:

   Newton/Euler for Rigid Bodies.
   Lagrange/Variational for Flexible Bodies.

2. Reference Frames:

   Both Kinematically and Dynamically Specified.

3. System Variables:

   Absolute Velocity for Dynamically Specified Sub-systems.
   Relative Velocity for Kinematically Specified Sub-systems.
   Generalized Momentum for Some Complex Sub-systems.
   Lagrange Multipliers for Closed Loops and Kinematic Constraints.

4. System Topology:

   Index Lists, Depth-First and Width-First Search Algorithms.

5. Treatment Constraints:

   Consistency Conditions for Kinematics and Closed Loops.
   Partitioning Algorithms for Parallel Computations.

6. Equation Generation:

   Numercial/Symbolic Calculations.

Recommended Computational Algorithms

1. Integrators:

   Semi-Explicit Methods for Rigid Bodies.
   Semi-Implicit Methods for Flexible Bodies.

2. Rotation Update:

   4-Parameter Euler Transformation.
   Euler-Rodrigue Rotation Matrix.

3. For Systems with Constraint Index ≥ 2:

   Special Equation Augmentation.
   Constraint Stabilization.

4. Provisions for Penalty Methods for Handling Constraint Equations.

4. Concurrent Computations:

   Partitioning Strategies.
   Software Considerations.
   Minimal Communications Algorithms.

N89-24667

CONSTRAINT ELIMINATION IN DYNAMICAL SYSTEMS

R. P. Singh and P. W. Likins*
DYNACS Engineering Company, Inc.
Clearwater, FL

THEME

Large Space Structures (LSS) and other dynamical systems of current interest are often extremely complex assemblies of rigid and flexible bodies subjected to kinematical constraints. This paper presents a formulation of the governing equations of constrained multibody systems via the application of singular value decomposition (SVD). The resulting equations of motion are shown to be of minimum dimension.

The motivation for this work was the development of a generic computer program for simulating space structures and similar electromechanical systems amenable to mathematical representation as a set of flexible bodies interconnected in a topological configuration. This representation may include closed loops of bodies, prescribed motion, or other constraints that may qualify as simple monholonomic. The equations of motion appropriate for a set of flexible bodies in an open loop configuration appear in Refs. 1, 2. A computer program (TREETOPS) developed to simulate the dynamic response of flexible structures in a topological tree configuration is described in Ref. 3. The SVD technique of the present paper is being incorporated in an extension of the TREETOPS program that permits application to constrained systems. This extension permits direct use of the dynamical equations for the less constrained system in Refs. 1, 2, with augmentation by kinematical constraint equations and reduction of dimension by SVD.

Basically, there are two conceptual approaches to solving the equations of motion of such systems. (1) One can introduce unknown forces and torques at the interfaces between constrained bodies (often accomplishing this symbolically with Lagrange multipliers), and then solve the dynamical equations simultaneously with the constraint equations to determine the constraint forces and torques as well as the kinematical variables, Ref. 4. (2) Alternatively, one can use the constraint equations to reduce the dimension of the system of dynamical equations to be solved by partitioning generalized coordinates, Refs. 5, 6. Techniques presented in Refs. 4, 5, 6 may encounter numerical singularities. Also, systems undergoing large motion may present problems of inconsistency in the constraints such as three-dimensional loops during the system motion becoming two dimensional or one-dimensional loops. In what follows, the SVD method will be shown to avoid mathematical singularities.

*Lehigh University, Bethlehem, PA.

Singular Value Decomposition: Orthogonal decomposition of an mxn matrix L by singular value decomposition is closely related to the eigenvalue-eigenvector decomposition of the symmetric positive semidefinite matrices L^TL and LL^T. Let r≤m be the rank of L. Then there are orthogonal matrices U and V of order mxm and nxn rexpectively such that

$$U^TLV = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} = S \tag{1}$$

where $\Sigma = \text{diag }(\lambda_1, \lambda_2, \ldots., \lambda_r)$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$.

The diagonal elements of the decomposition are called the singular values of the matrix L. The singular values are unique, although U and V are not.

It is easy to verify that

$$V^TL^TLV = \text{diag }(\Sigma^2, 0) \tag{2}$$

Thus $(\lambda_1^2, \ldots., \lambda_r{}^2)$ must be the nonzero eigenvalues of L^TL arranged in the descending order and the requirement that λ_i be nonnegative completely determines the λ_i. The eigenvectors of L^TL are the columns of V. If L^TL has a multiple eigenvalue $\lambda^2 > 0$, the corresponding columns of V may be chosen as an orthonormal basis for the space spanned by the eigenvectors corresponding to λ^2.

From eq. (1)

$$L = USV^T \tag{3}$$

Now with proper partitioning of U and V eq. (3) can be expressed as

$$L = \begin{bmatrix} U_1 \vdots U_2 \end{bmatrix} \left[\begin{array}{c:c} \Sigma & 0 \\ \hdashline 0 & 0 \end{array}\right] \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma V_1^T \tag{4}$$

From the above one obtains

$$U_1 = LV_1 \Sigma^{-1} \tag{5}$$

Thus once V_1 is chosen U_1 is obtained by eq. (5). The matrices U_2 and V_2 may be any matrices with orthonormal columns spanning the null spaces of L^T and L, respectively. It is worthwhile to mention that the null space of L is the space of all vectors x such that

$$Lx = 0 \tag{6}$$

With the orthogonal decomposition given by eq. (3), an nxm matrix L^+, called the pseudoinverse of L, is defined by

$$L^+ = V \left[\begin{array}{c|c} \Sigma^{-1} & 0 \\ \hline 0 & 0 \end{array}\right] U^T \tag{7}$$

L^+ is uniquely defined by L; it does not depend on the particular orthogonal decomposition of L.

Application of SVD to Dynamical System with Constraints: Let $q = q_1, \ldots, q_n$ comprise a set of generalized coordinates that fully defines the configuration of the dynamical system. The equations of motion of the system can be written as

$$M\ddot{q} = F(\dot{q},q,t) \tag{8a}$$

where the elements of nxn matrix M are functions of q's and the inertia properties of the system; the elements of nx1 column vector F are functions of q's, their time derivatives $\dot{q}$'s and applied forces (moments) on the systems. If the generalized coordinates are related by constraint equations then they are not independent and the right hand side of eq. (8a) will also include the nonworking forces of constraints. Let the unknown constraint forces be denoted F^c. Now for the general case of constrained dynamical system, eq. (8a) takes the following form

$$M\ddot{q} = F + F^c \tag{8}$$

Suppose however that the constraint equations can be written as

$$A\dot{q} = B \tag{9}$$

where A is of dimension mxn (m<n) and B is an mx1 column vector.

Holonomic constraint equations can always be placed in the form of eq. (9) and nonholonomic constraints in the class called Pfaffian or simple have this structure also.

If the rank of matrix A is r≤m then r of the kinematical variables in q are related by eq. (9) and there are only n-r independent generalized coordinates. In other words the dynamical system possesses n-r degrees of freedom.

The SVD of the mxn matrix A provides

$$A = USV^T \tag{10}$$

The orthogonal matrices U and V (of dimension mxm and nxn, respectively) are partitioned as

$$U = \left[U_1 \,\vdots\, U_2 \right] \qquad (11)$$

$$V = \left[V_1 \,\vdots\, V_2 \right] \qquad (12)$$

where U_1 and V_1 are respectively mxr and nxr matrices; U_2 and V_2 are respectively mx(m-r) and nx(n-r) matrices. Note that r is the rank of A.

Because $AV_2 = 0$, eq. (9) is satisfied by

$$\dot{q} = A^+B + V_2\dot{z} \qquad (13)$$

for any vector $\dot{z}$, A^+ is the pseudoinverse of A. We shall refer to z as the reduced set of (n-r) coordinates.

Differentiation of eq. (9) with respect to time yields

$$A\ddot{q} = -\dot{A}\dot{q}+\dot{B}$$

$$\text{or, } A\ddot{q} = B' \qquad (14)$$

Following eq. (13) express $\ddot{q}$ in terms of $\ddot{z}$ as

$$\ddot{q} = A^+B' + V_2\ddot{z} \qquad (15)$$

Note from eq. (13) or eq. (15) that V_2 maps the n kinematic variables $\dot{q}$ (or $\ddot{q}$) to n-r variables $\dot{z}$ (or $\ddot{z}$). Thus a consistent set of equations of motion in $\ddot{z}$ is given as

$$V_2^TMV_2\ddot{z} = V_2^TF + V_2^T F^c - V_2^T MA^+B' \qquad (16)$$

The coefficient of $\ddot{z}$ is a symmetric positive definite matrix with the characteristic of an "inertia matrix" for the reduced set of coordinates z.

With the Lagrange multiplier method, F^c is established via (see Ref. 4)

$$F^c = A^T\alpha \qquad (17)$$

where α is the column vector of Lagrange multipliers.

Premultiply eq. (17) by V_2^T to obtain the following

$$\begin{aligned} V_2^TF^c &= V_2^TA^T\alpha \qquad (18) \\ &= (AV_2)^T\alpha \\ &= 0 \end{aligned}$$

Thus it is seen that the nonworking constraint forces make no contribution to the equations of motion (eq. (16)) and need not be recorded.

Employing the transformations given by eqs. (13) and (15), the minimum dimension governing differential equations of motion are given by

$$V_2^T M V_2 \ddot{z} = V_2^T F - V_2^T M A^+ B' \tag{19}$$

and

$$\dot{q} = A^+ B + V_2 \dot{z} \tag{20}$$

This method eliminates the forces of constraints which when included serve not only to enlarge the dimension of the dynamical system but also quite often introduce computational problems.

ACKNOWLEDGMENT

An excellent treatment of the computational efficiency of the SVD is given in Ref. 7.

REFERENCES

1. Singh, R. P. and Likins, P. W., "Manipulator Interactive Design with Interconnected Flexible Elements", Proceedings of the 1983 Automatic Control Conference, San Francisco, CA., June 1983, pp. 505-512.

2. Singh, R. P., VanderVoort, R. J. and Likins, P. W., "Dynamics of Flexible Bodies in Tree Topology - A Computer Oriented Approach", Proceedings of the 25th SDM and AIAA Dynamics Specialist Conference, Palm Springs, CA., May 1984, Part 2, pp. 327-337.

3. VanderVoort, R. J. and Singh, R. P., "A Pointing Control System Simulation and Generic Model Generator for Multi-Body Flexible Structures" Proceedings of the Ninth DARPA Strategic and Space Symposium, Monterey, CA., October 1983.

4. Bodley, C., Devers, A., Park A. and Frisch, H., "A Digital Computer Program for Dynamic Interaction Simulation of Controls and Structures (DISCOS)", NASA Technical Paper 1219, May 1978.

5. Kane, T. R,. and Wang, C. F., "On·the Derivation of Equations of Motion", J. SIAM, Vol. 13, 1965, pp. 487 - 492.

6. Wehage, R. A. and Haug, E. J., "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems", ASME Journal of Mechanical Design, Vol. 104, January 1982, pp. 247-255.

7. Juang, Jer-Nan and Pinson, Larry D., "Application of Singular Value Decomposition to Structural Dynamics Systems with Constraints," AIAA Paper No. 85-0687-CP, April 1985.

N89-24668

NONLINEAR CHARACTERISTICS OF JOINTS AS ELEMENTS OF MULTI-BODY DYNAMIC SYSTEMS

Edward F. Crawley
Space Systems Laboratory
Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Mass.

Introduction

As the connecting elements in Multi-Body structures, joints play a pivotal role in the overall dynamic response of these systems. Obviously, the linear stiffness of the joint strongly influences the system frequencies, but the joints are also likely to be the dominant sources of damping and nonlinearities, especially in aircraft and space structures. The general characteristics of such joints will be discussed. Then the state of the art in nonlinear joint characterization techniques will be surveyed. Finally, the impact that joints have on the overall response of structures will be evaluated.

Although somewhat difficult to assess, the rough order of magnitude of various dissipative mechanisms is shown (based on critical damping equaling unity). In Earth-based structures, transmission losses probably dominate. But in aeronautical structures, dissipation in joints begins to become more important. In space, in the absence of transmission losses, joints dominate the passive dissipation mechanisms.

Order of Magnitude of Structural Dissipative Mechanisms

	Earth	Aero	Space
Support transmission	10^{-1}	0	0
Aeroacoustic transmission	10^{-2}	10^{-2}	0
Material damping	10^{-3}	10^{-3}	10^{-3}
Joint damping	10^{-2}	10^{-2}	10^{-2}
Active control	10^{-1}	10^{-2}	10^{-2}

The potential nonlinear characteristics of a space structure are compared with the stiffness (normalized to unity). In the absence of yielding, material nonlinearities will be on the order of fractions of a percent. Geometric large deflection, at least in the flexible modes, is small. Therefore the strong nonlinearities of the joints are again likely to dominate.

ORDER OF MAGNITUDE OF SPACE STRUCTURAL NONLINEARITIES

MATERIAL STIFFNESS	10^{-3}
MATERIAL DAMPING	10^{-3}
GEOMETRIC LARGE DEFLECTION	10^{-2}
GEOMETRIC JOINT NONLINEARITY	10^{-1}

THEREFORE JOINTS ARE THE LARGEST SOURCE OF PASSIVE DAMPING AND NONLINEARITY.

The overall characteristics of material damping, listed below, coupled with the fact that the material damping is likely to be one-half to one order of magnitude less than joint damping, tend to make this a relatively less critical area in modeling.

MATERIAL DAMPING CHARACTERISTICS

- DISTRIBUTED WITH STIFFNESS, THEREFORE MODAL DAMPING IS PROPORTIONAL, MODES ARE REAL AND UNCOUPLED
- ONLY WEAKLY NONLINEAR, THEREFORE APPROXIMATE, MODELS ARE SUFFICIENTLY ACCURATE
- HAS ORIGINS IN REASONABLY WELL-UNDERSTOOD MECHANISMS, E.G., THERMAL TRANSPORT, PLASTICITY
- IS DEPENDENT ON GLOBAL FREQUENCY, AMPLITUDE TEMPERATURE AND HUMIDITY ENVIRONMENT

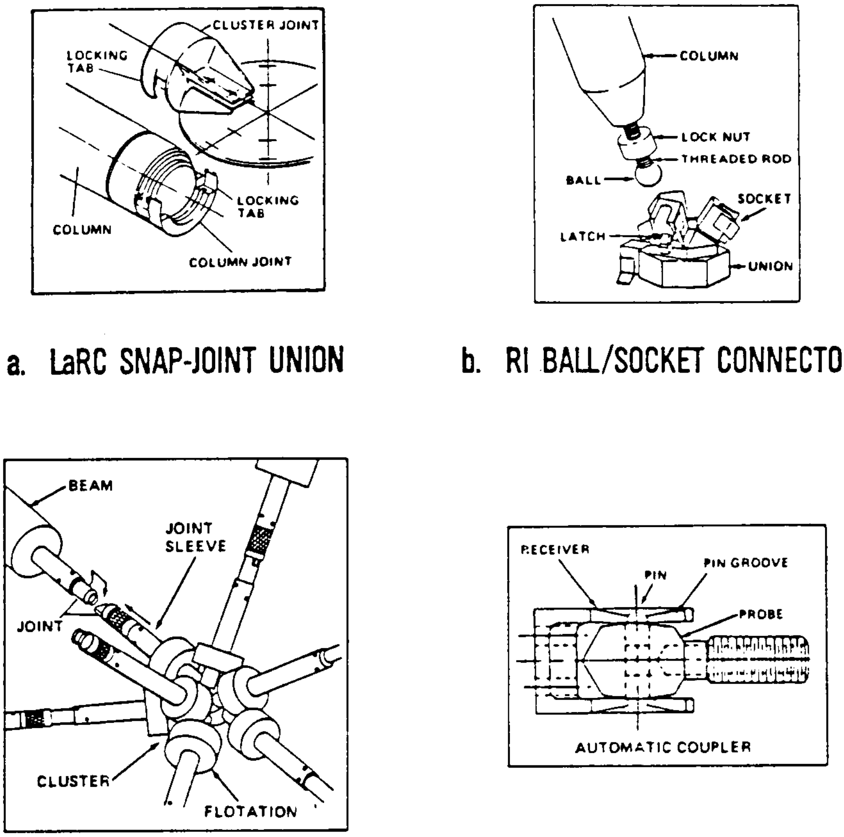
The corresponding characteristics of joints, their discrete locations, strongly nonlinear behavior, and somewhat obscure micromechanics, make this a more challenging area for modeling. Despite frequent attempts in the history of aerospace technological development, no unified analysis approach to this modeling has been developed.

JOINT CHARACTERISTICS

- NOT DISTRIBUTED, BUT OCCUR AT DISCRETE LOCATIONS, THEREFORE MODAL DAMPING IS NOT PROPORTIONAL, AND MODES ARE LINEARLY COUPLED AND COMPLEX
- STRONGLY NONLINEAR, THEREFORE MODES STIFFEN AND COUPLE NONLINEARLY
- HAS ORIGINS IN RELATIVELY POORLY UNDERSTOOD MECHANISMS, E.G., MICROSLIP FRICTION, IMPACTING

To gain some insight into this difficulty, it is useful to look at several proposed joint geometries for deployable space structures. Note that the geometries are all quite different, but all have several characteristics in common. There must be some amount of play in the joint to allow for assembly but some stiffening or locking mechanism to make the joint fixed when deployed. This combination of play and fixity leads to the impacting and nonlinear stiffness typical of such joints.

TYPICAL JOINT DESIGNS FOR DEPLOYED SPACE STRUCTURES



a. LaRC SNAP-JOINT UNION

b. RI BALL/SOCKET CONNECTOR

c. MIT CLUSTER SLIP-JOINT

d. VOUGHT QUICK-CONNECT COUPLER

Not only do the properties of joints depend on the overall geometry of their design, but these properties depend on a number of details. The surface of the contacting elements can depend, for example, on the quality of machining, the load and wear history, and the duration on orbit. Even nominally identical joints can have a statistical variation due to manufacturing tolerance. Therefore, in realistic assemblies, direct calculation of properties is somewhat unproductive.

- JOINT PROPERTIES DEPEND ON VERY LOCAL DETAILS
  1. SURFACE FINISH, LUBRICATION, OUTGASSING AND OXIDATION
  2. WEAR AND TRIBOLOGY
  3. PRECISION OF FIT AND ALIGNMENT
  4. PRELOAD AND INITIAL DEFORMATION
  5. LOCAL THERMAL DEFORMATIONS

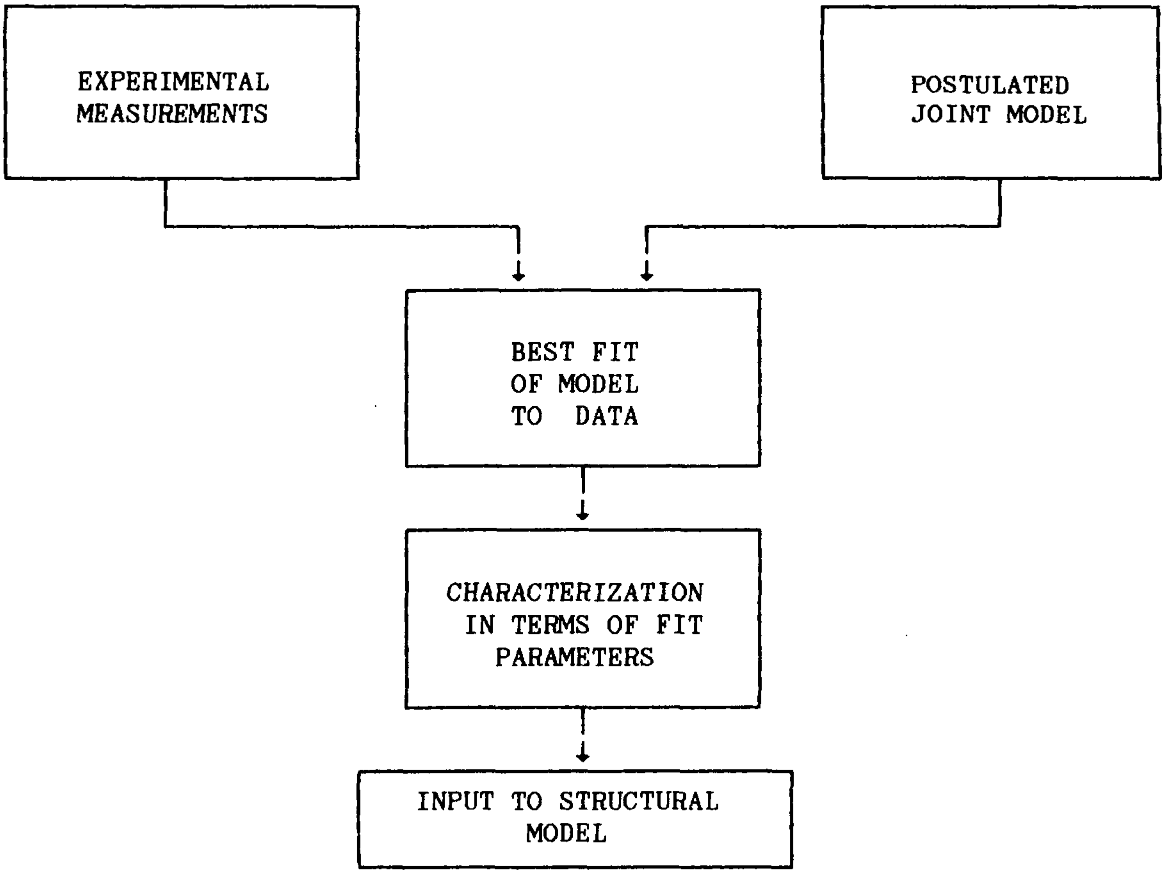
- JOINTS OF IDENTICAL MATERIALS CAN HAVE VERY DIFFERENT BEHAVIOR

- NOMINALLY IDENTICAL JOINTS MAY HAVE A STATISTICAL VARIATION IN BEHAVIOR

THEREFORE THE DETAILED CALCULATION OF JOINT CHARACTERISTICS FROM FIRST PRINCIPLES IS UNPRODUCTIVE.

A more common approach to characterization is a hybrid of simplified modeling and experimentation. A set of experiments is run, yielding some data on the force transmission of the joint. Concurrently, several postulated models of the joint are developed. Often this is somewhat interactive, i.e., after the data are evaluated, refined models are postulated. The force characteristics, or structural response of the postulated model, is then compared with the experimental data, and some fit of the model to data is performed. Based on this fit, the parameters of the model are available for use in the overall structural model.

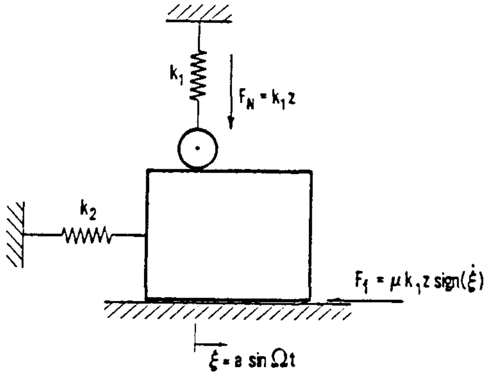
ALTERNATIVE TO DIRECT CALCULATION


EXPERIMENTAL MEASUREMENTS
POSTULATED JOINT MODEL
BEST FIT OF MODEL TO DATA
CHARACTERIZATION IN TERMS OF FIT PARAMETERS
INPUT TO STRUCTURAL MODEL

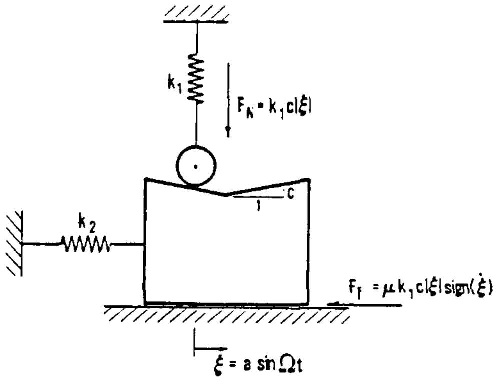
A number of different models of joint behavior can be postulated. Three of the common ones are shown below. The first is Coulomb friction, in which the normal force, and therefore the frictional drag force, remains constant. In displacement dependent friction, the normal and frictional drag forces vary with displacement. This model is probably more realistic for jointed trusses in the absence of thermal and gravity loads. When the deadband closes, impacting occurs, and a sharp jump in damping and stiffness (not shown) occurs.

EXAMPLE POSTULATED PIECEWISE LINEAR MODELS

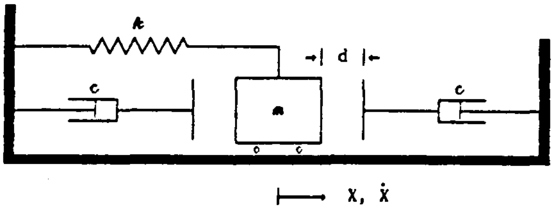
COULOMB FRICTION



DISPLACEMENT DEPENDENT FRICTION [1]



IMPACTING



The principal characteristics of four procedures for identification of nonlinear elements are shown. The first two are extensions of techniques developed for linear systems and are more easily extendable to multi-dof-models. However, they are probably only appropriate for weak nonlinearities. The latter two are currently limited to single-dof systems, but can handle stronger nonlinearities. A more detailed explanation of each will follow.

EXPERIMENTAL IDENTIFICATION PROCEDURES

PROCEDURE	MEASUREMENT	DOMAIN OF FIT	NONLINEARITY	DOF
FREQUENCY DOMAIN MODAL IDENTIFICATION [2] (EWINS,IMP.COL.)	F, $\ddot{X}$ VS ω	FREQUENCY	WEAK	SEVERAL
TRANSIENT TIME DOMAIN [3] (HORTA, JUANG, LARC)	F, $\ddot{X}$ VS T	TIME	WEAK	SEVERAL?
CLASSICAL FORCE- DISPLACEMENT [4] (SONI, UDRI)	F VS X	PARTIAL STATE SPACE	STRONG	ONE
FORCE-STATE [5] (CRAWLEY, MIT)	F VS X, $\dot{X}$	STATE SPACE	STRONG	ONE

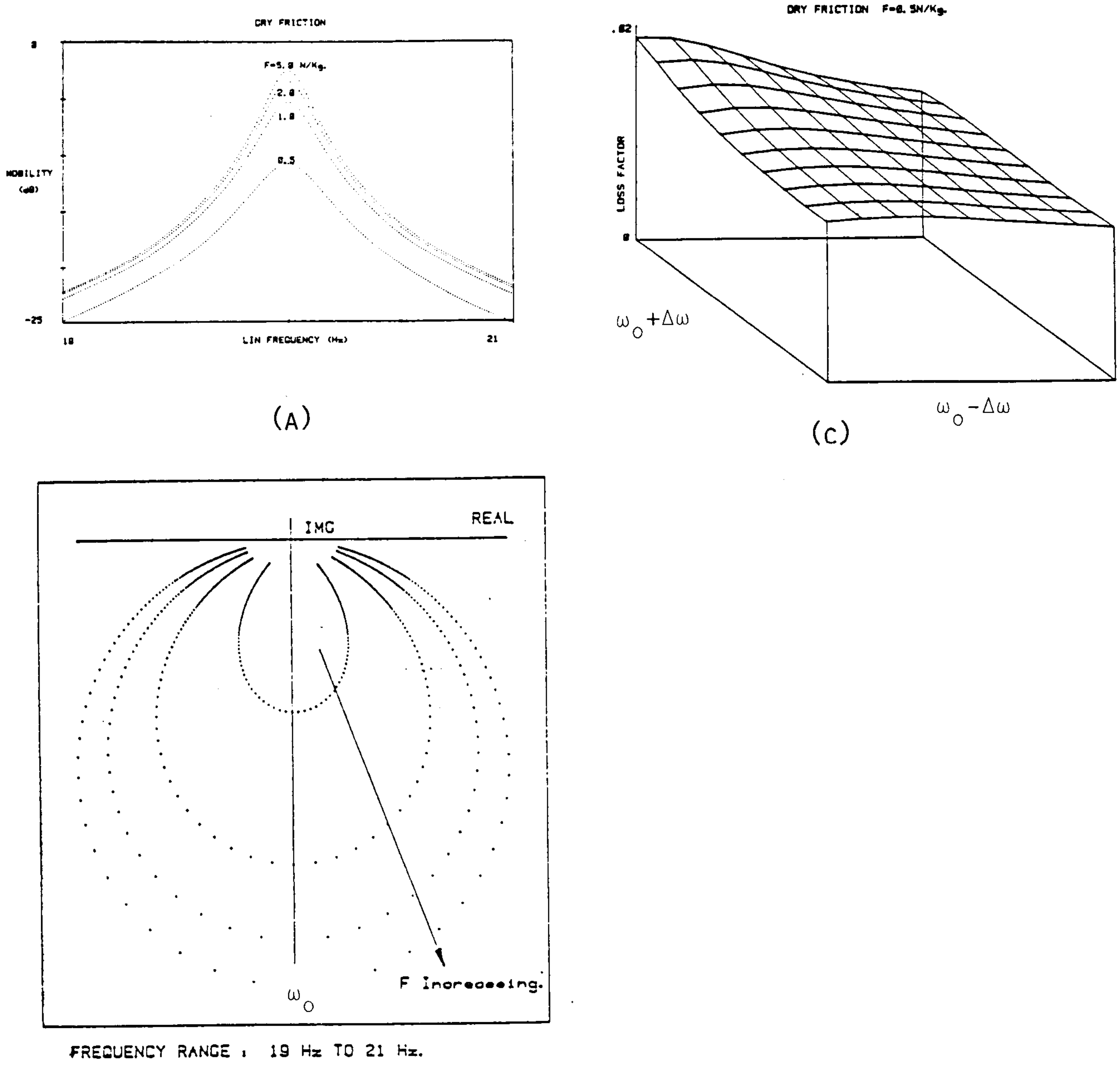
The extended frequency domain modal identification procedure was developed simply to uncover the presence of nonlinearities in modal test data. Therefore, the output is limited to indications of the presence, strength and type of nonlinearity. It is best used as a diagnostic tool in checking the consistency of test data.

EXTENDED FREQUENCY DOMAIN MODAL PROCEDURE [2]

- POSTULATE NONLINEARITY AND CALCULATE LOSS FACTOR USING NYQUIST PLANE RESPONSE
- MEASURE RESPONSE AT RESONANCE AND CALCULATE LOSS FACTOR
- IF LOSS FACTOR IS INCONSISTENT (I.E., NOT CONSTANT), CHOOSE POSTULATED NONLINEARITY WHICH BEST FITS OBSERVED BEHAVIOR
- OUTPUT - APPROXIMATE INDICATION OF TYPE AND DEGREE OF NONLINEARITY IN MODAL RESONANCE.

The technique is a direct extension of the procedure for extracting frequency and loss factor parameters from transfer functions, as presented in the Nyquist plane (b). A simple, single-dof response appears as a perfect circle in this representation, tangent to the real axis at the origin. Any deviation from this circle is due to a nonlinearity (or presence of multiple poles). The loss factor (damping ratio) can be calculated by choosing pairs of points about ω_o, forming a matrix of computed values. Figure (c) is a graphical representation of loss factor calculated on this matrix. For a linear system, this surface would be flat. The shape shown is typical of a system with Coulomb friction.

CALCULATED FREQUENCY RESPONSE (A) AND NYQUIST REPRESENTATION (B) FOR A SDOF SYSTEM WITH COULOMB FRICTION. THE LOSS FACTOR (C) IS INFERRED FOR A RANGE OF FREQUENCY SPREAD ABOUT THE RESONANCE.



(A)

(C)

(B)

Likewise, the existing time domain techniques are extensions of techniques developed for linear systems. These techniques generally examine the transient free response to extract system mode shapes and frequencies. Weak nonlinearities appear as a frequency with a number of higher harmonics. Each type of nonlinearity has such a signature.

EXTENDED EIGENSYSTEM REALIZATION ALGORITHM [3]

- POSTULATE NONLINEARITY AND CALCULATE FOURIER CONTENT OF TRANSIENT FREE RESPONSE

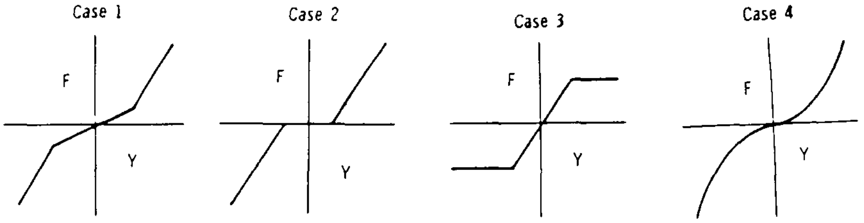
- MEASURE FREE RESPONSE AND IDENTIFY FOURIER CONTENT WITH ERA

- COMPARE MEASURED HIGHER HARMONIC CONTENT OF OF MODAL RESPONSE WITH SIGNATURES OF POSTULATED NONLINEARITIES

- OUTPUT - APPROXIMATE INDICATION OF TYPE AND DEGREE OF NONLINEARITY IN RESPONSE.

Four example cases are shown, all typical of a stiffening or softening spring. The Fourier components of the free response of such a spring in a spring mass system were calculated. The calculated response was also fed as simulated data to the ERA program and the harmonics of the response calculated. Good capability to reconstruct the signature of a known nonlinearity is shown. However, the recognition of the signature of an unknown nonlinearity is still under development.

FOUR GENERIC NONLINEAR JOINTS (A) AND THE FOURIER CONTENT OF THEIR TRANSIENT DECAY FROM ANALYSIS AND ERA IDENTIFICATION OF COMPUTED RESPONSE (B).



(A)

	FREQUENCY Hz.		COMPONENT AMP.	
CASE NO.	ERA	ANALYSIS	ERA	ANALYSIS
1	0.135	0.134	0.986	0.987
	0.404	0.403	0.015	0.015
	0.673	0.672	-0.002	-0.002
2	0.125	0.125	0.979	0.979
	0.375	0.375	0.023	0.023
	0.625	0.625	-0.003	-0.003
3	0.096	0.096	1.030	1.030
	0.289	0.289	-0.035	-0.035
	0.482	0.482	0.006	0.006
4	0.456	0.456	0.961	0.961
	1.369	1.369	0.038	0.038
	2.281	2.281	0.001	0.001

(B)

The classical approach to the problem is, of course, to simply put a joint in a testing machine and develop force-stroke data, as shown on the next page. From such data secant modulus and average loss factor can be calculated. The limitation is that such properties are already smeared, or averaged over the stroke, and no tangent or point properties can be determined. Further, the dependence of the force on the rate of change of stroke is lost.

QUASI-STEADY FORCE DEFLECTION PROCEDURE [4]

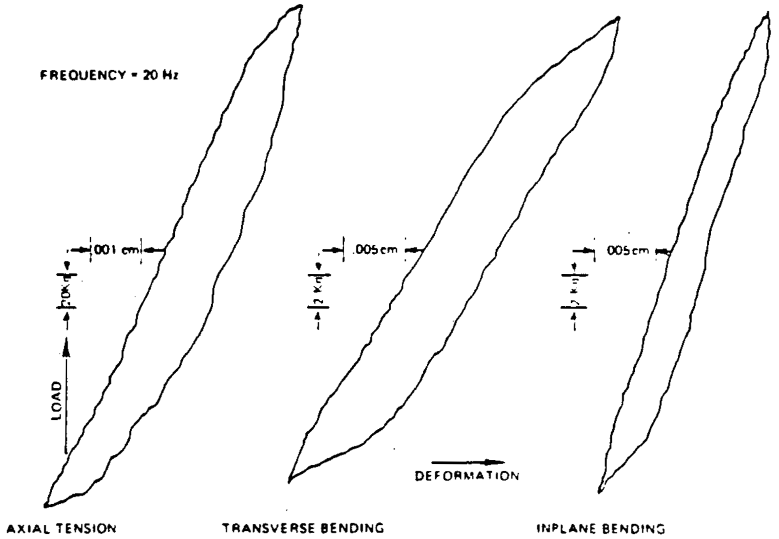
- POSTULATE NONLINEARITY AND CALCULATE ITS F VS X BEHAVIOR

- MEASURE F VS X BEHAVIOR FOR ONE X_{MAX} AND ω, AND CALCULATE EFFECTIVE STIFFNESS AND LOSS FACTOR

- REPEAT AT DIFFERENT AMPLITUDES AND FREQUENCIES

- OUTPUT - EFFECTIVE STIFFNESS AND LOSS AS A FUNCTION OF FREQUENCY AND AMPLITUDE.

TYPICAL FORCE-STROKE CYCLES OF THE THREE STRESS RESULTANTS OF A JOINT AT ONE LOAD AND FREQUENCY.



The force-state mapping procedure is designed specifically to identify strong nonlinearities in joints and addresses the two limitations of the classic Force-Stroke measurement. The dependence of transmitted force on both displacement and velocity is explicitly determined, and local or tangent values produced. At the current time the procedure is limited to single-dof systems.

FORCE-STATE MAPPING PROCEDURE [5]

- POSTULATE NONLINEARITY AND CALCULATE F VS X, $\dot{X}$ BEHAVIOR
- MEASURE F VS X, $\overset{\prime}{X}$ OVER EXPECTED RANGE
- FIT POSTULATED SURFACES TO DATE IN FORCE-STATE SPACE
- OUTPUT:
  1. RAW DATA FOR DATA LOOK-UP
  2. GLOBAL FIT PARAMETERS FOR ANALYTIC REPRESENTATION
  3. LOCAL EQUIVALENT SECANT MODULI FOR LINEARIZED REPRESENTATION

The force-state maps of three simple systems are shown. A spring (a) produces a plane with a slope against x, but no change in $\dot{x}$. A linear viscous damper would produce a plane with a slope against x, and no change in x. Thus any linear element (i.e., spring and damper) will have a map which is a flat plane in force-state space. Any deviation from a plane is indicative of a nonlinearity.

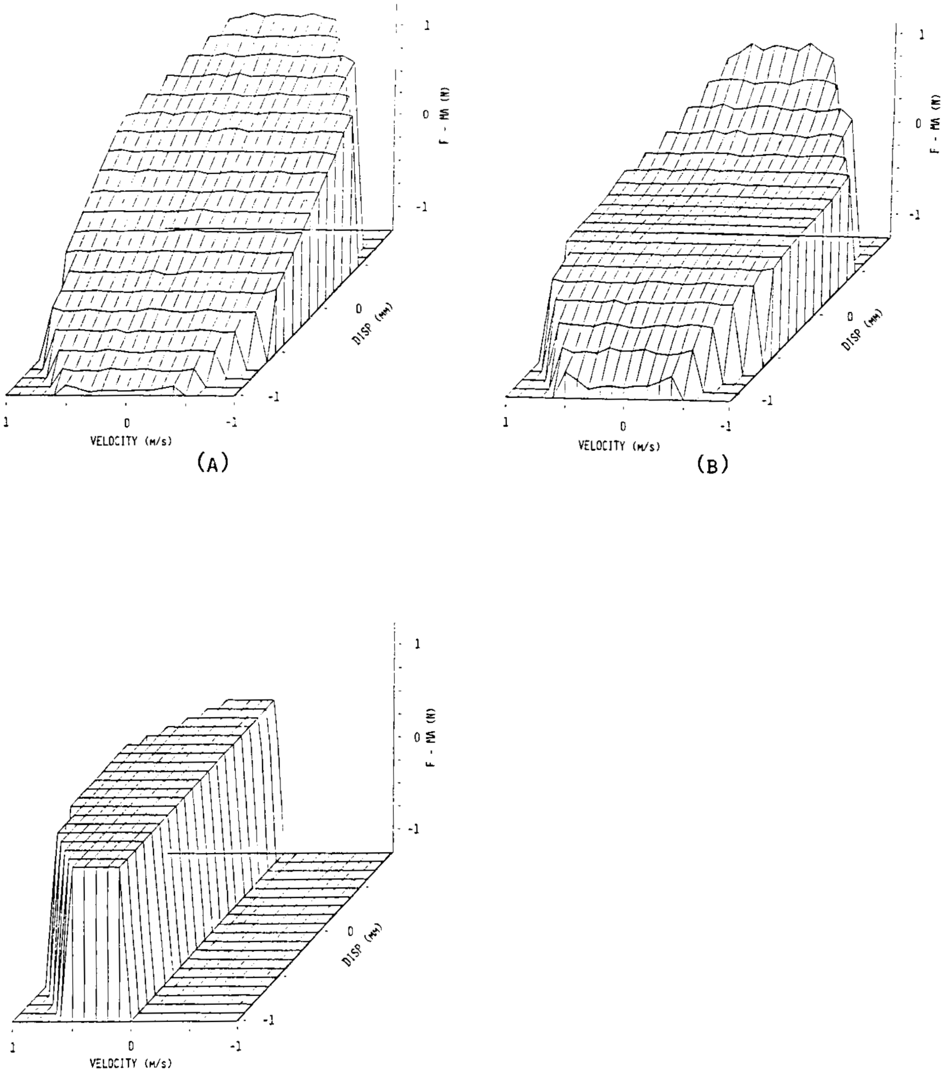
Two common nonlinearities are shown in (b) and (c). The cubic spring nature is clear in fig. (b). Figure (c) shows the map of Coulomb friction, which is independent of x, and takes on the sign of the velocity.

FORCE-STATE MAPS OF:

A) LINEAR SPRING

B) CUBIC SPRING

C) COULOMB FRICTION



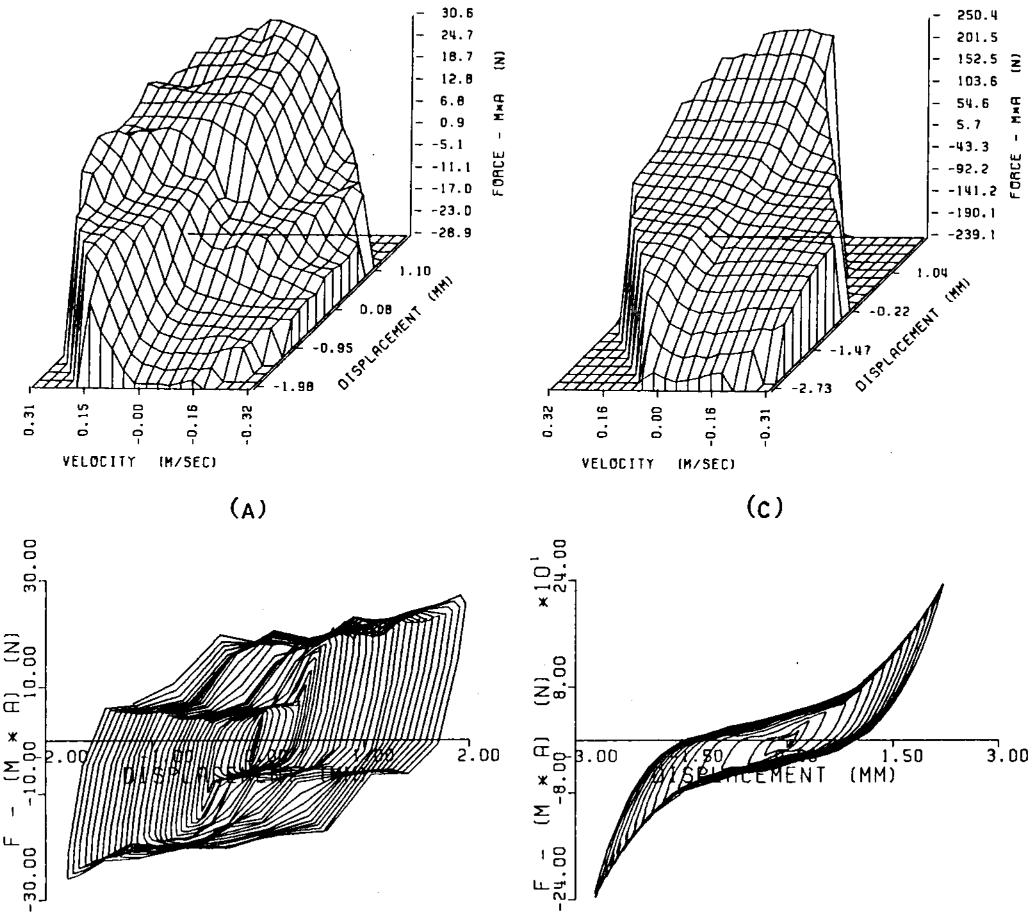
(A)

(B)

(C)

The force-state maps of a real joint are shown below. The joint is of a quick-disconnect-pin and clevis type, similar to the Vought connector shown earlier. The figures on the left (a and b) show the characteristic without a stiffening sleeve. Note the step at $\dot{x}$ equals zero, indicative of friction. With the addition of a sleeve, the joint becomes stiffer (note the change in vertical scale), and the cubic stiffness of the sleeve begins to dominate. A strong dissipative nature is still obvious from the hysteresis loops in fig. (d).

THE FORCE-STATE AND FORCE-DISPLACEMENT CURVES OF A CLEVIS-PIN-TYPE CONNECTOR WITHOUT (A AND B) AND WITH (C AND D) A REINFORCING SLEEVE.



(A)

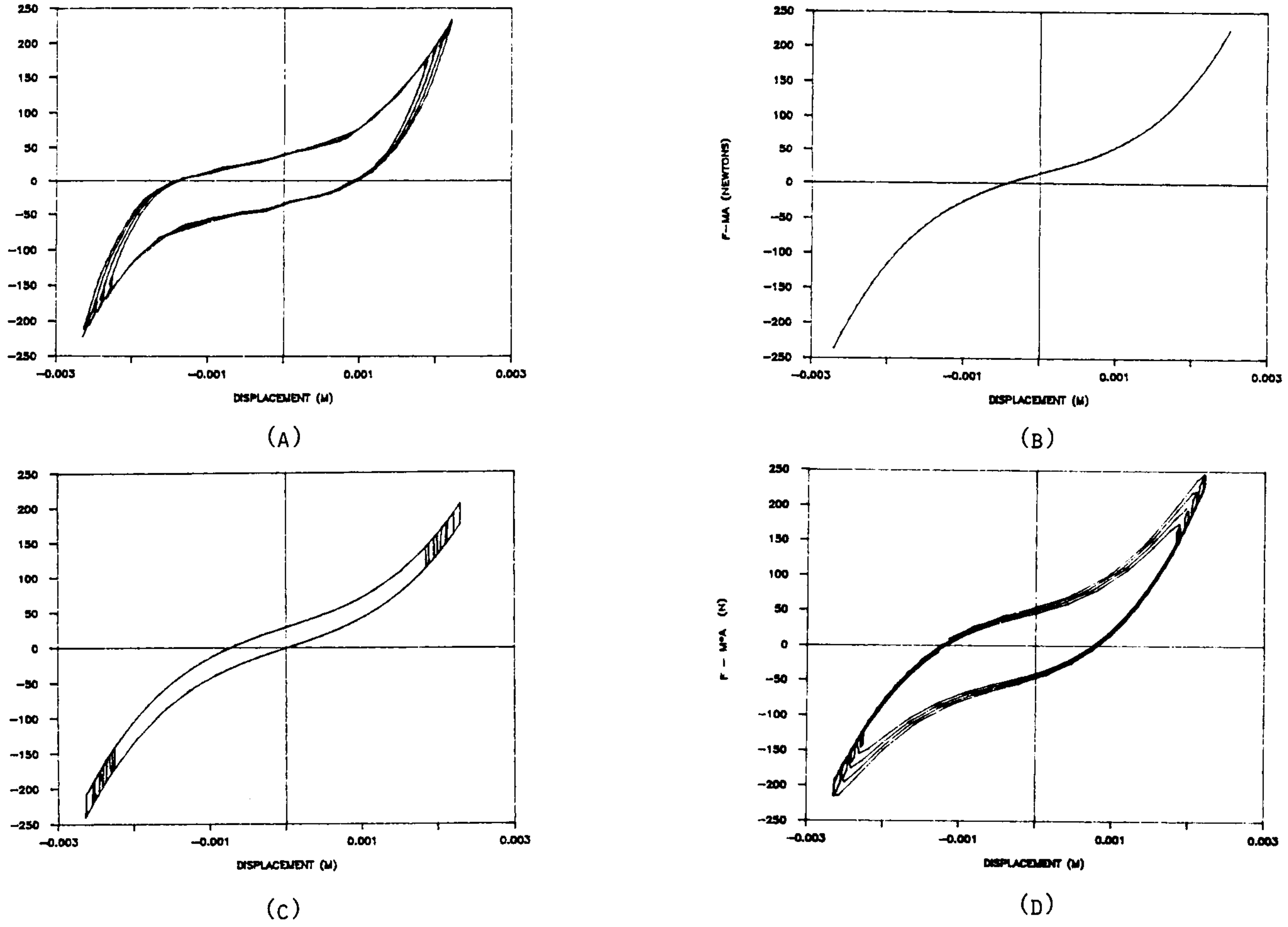
(C)

(B)

(D)

In an effort to fit a postulated model to the data on the previous page (i.e., the joint with sleeve), three successively refined models were used. In fig. (b), a cubic, plus linear, spring term approximately matches the shape but, of course, has no dissipative nature. The addition of friction introduces the classic hysterectic step. Finally, with the introduction of linear damping, the measured data are closely reproduced.

SUCCESSIVE APPROXIMATIONS OF ACTUAL DATA (A) BY A CUBIC SPRING (B), CUBIC SPRING, PLUS FRICTION (C) AND CUBIC SPRING, FRICTION AND LINEAR DAMPING (D).



(A)

(B)

(C)

(D)

The requirements for efficient computation place several requirements on the identification scheme. It is highly desirable to have available the force-state information for direct pseudo-force computation.

COMPUTATIONAL CONSIDERATIONS [6]

- THREE COMPUTATIONAL APPROACHES TO INCLUDING THE JOINT NONLINEARITY CAN BE CONSIDERED

  1. HOMOGENEOUS NONLINEARITY, EXPLICIT OPERATOR
  2. HOMOGENEOUS NONLINEARITY, IMPLICIT OPERATOR
  3. PSEUDO-FORCE REPRESENTATION [7]

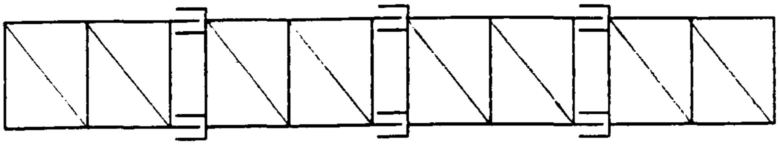
- IN ALL THREE, BUT ESPECIALLY IN THE PSEUDO-FORCE METHOD, IT IS NECESSARY TO HAVE THE JOINT CHARACTERISTIC IN TERMS OF JOINT STATE VARIABLES.

- IF ONLY AVERAGE, OR SECANT PROPERTIES ARE KNOWN, THEN CONSIDERABLE ITERATION IS REQUIRED, AND TRANSIENT ANALYSIS MAY NOT BE ACCURATE.

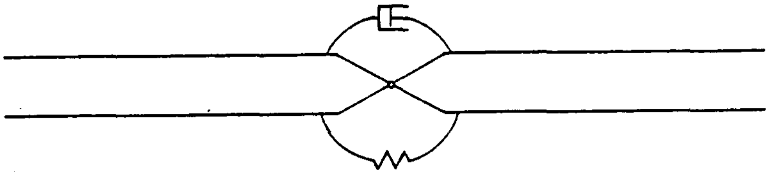
As an example problem, a four-bay truss, connected by joints, is modeled as a four segment beam, pinned in translation. In rotation it is constrained by a linear spring and damper.

EFFECTS OF JOINTS ON MODAL PROPERTIES

MODEL A CONNECTED 2-D TRUSS

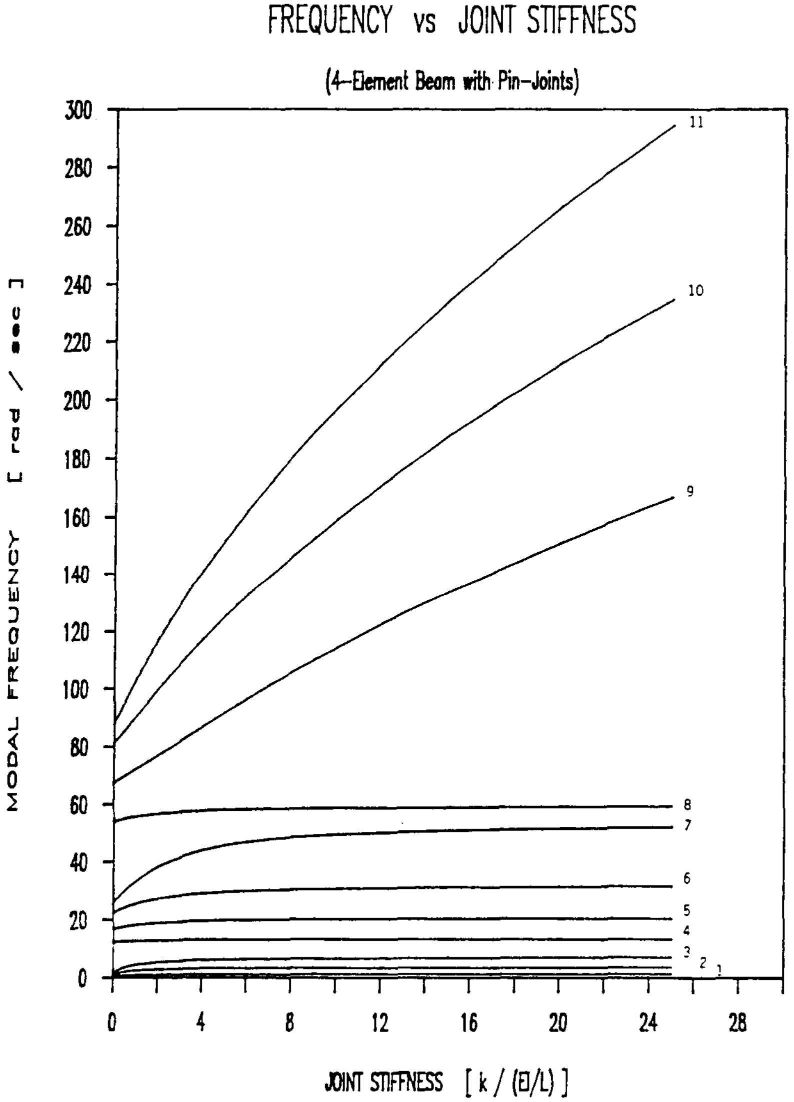


AS A PINNED BEAM OF 4 ELEMENTS WITH ROTARY SPRINGS AND DAMPERS.



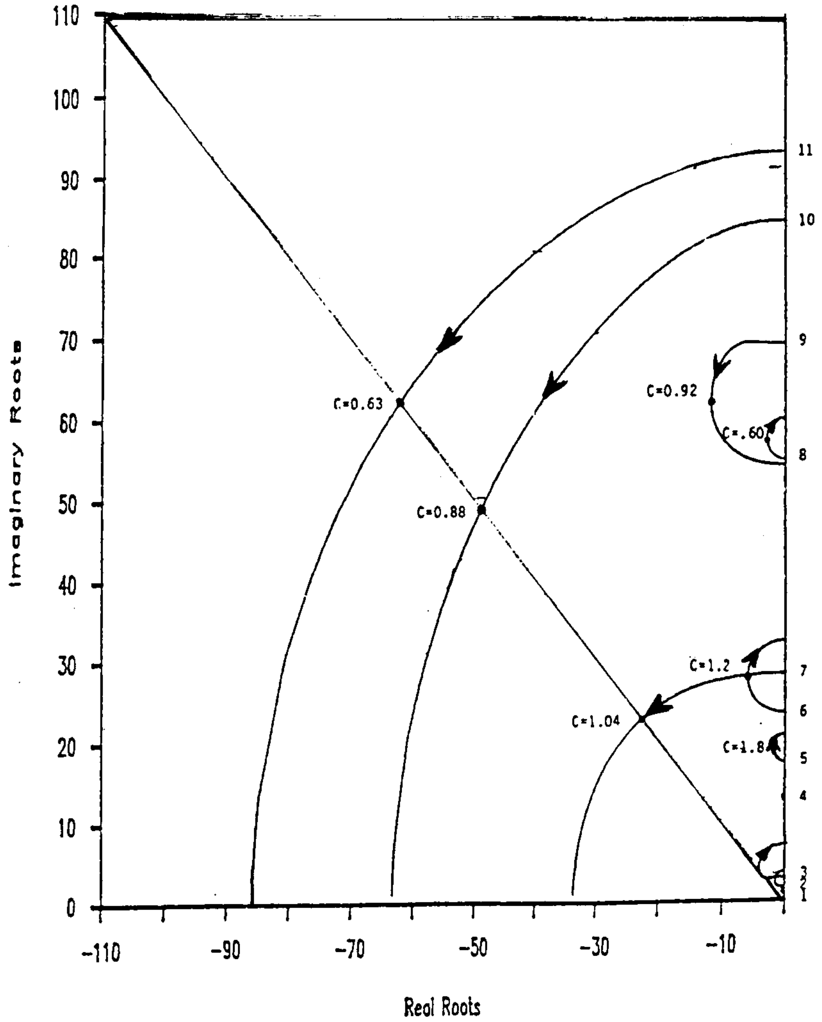
When the natural frequencies of the system are plotted versus nondimensional joint stiffness, their trends are apparent. Of course, all modes stiffen as k is increased. Some modes, such as #4, are only slightly affected, while others, such as #7, are strongly affected. The lowest eight modes are asymptotic to a constant frequency, while the highest three continue to rise as k increases.

STIFFENING EFFECTS OF JOINTS AS A FUNCTION OF K_{JOINT}.



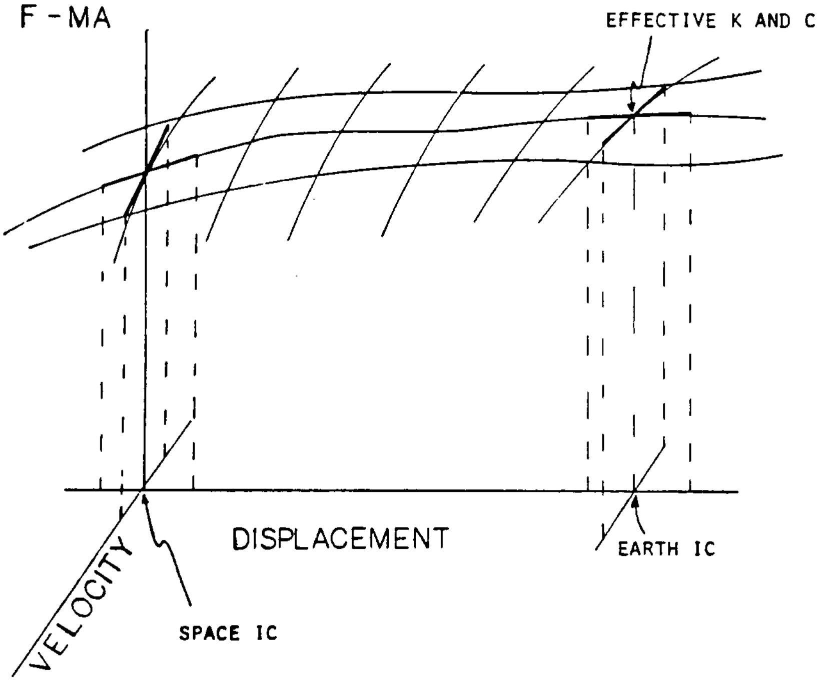
The addition of linear joint damping has some surprising results. Note that in only three modes, 7, 10 and 11, is the damping roughly proportional, i.e., the pole is driven to the real axis. In most modes, the root damps, then asymptotically stiffens and loses damping. In one mode, #9, the frequency drops.

LOCUS OF ROOTS FOR INCREASING LINEAR JOINT DAMPING, FOR $K_{JOINT} = 0.3\ EI/\ell$



Finally, this figure shows an interesting application of the force-state map to Earth testing of space structures. Suppose a structure was suspended in one gravity in such a way that the gravity load caused a steady deflection. The small displacement vibration would then take place about this "Earth IC," and would have the effective stiffness and damping shown. In space, in the absence of gravity loads, there would be no steady deflection and the effective K and C would be about a "Space IC," as shown. For a generally nonlinear joint, these properties could be completely different from those of the Earth test, leading to differences in dynamic behavior on orbit when compared to those measured on Earth.

USE OF THE FORCE-STATE MAP TO DETERMINE THE EFFECTIVE STIFFNESS AND DAMPING IN A JOINTED STRUCTURE, AS WOULD BE MEASURED ON EARTH AND IN SPACE.



SUMMARY

- DETAILED MODELING OF MICROMECHANICS OF THE JOINT NOT PRODUCTIVE
- DEVELOPMENT OF SIMPLE GENERIC MODELS USEFUL
- IMPROVED NONLINEAR IDENTIFICATION NECESSARY.

References

1. Hertz, T.J., and Crawley, E.F., "Displacement Dependent Friction in Space Structural Joints," AIAA Journal, vol. 23, Dec. 1985, pp. 1998-2000.

2. Ewins, D.J., and Sidhu, H., "Modal Testing and the Linearity of Structures," Mecanique Materiaux Electricite, No. 389-390-391, Mai-Juin-Juillet 1982.

3. Hortz, L.G., Juang, J.-N., "Identifying Approximate Linear Models for Simple Nonlinear Systems," AIAA Paper 85-0686, 26th Structures, Structural Dynamics and Materials Conference, Orlando, Florida, April 1985.

4. Soni, M.L., and Agrawal, B.N., "Damping Synthesis for Flexible Space Structures Using Combined Experimental and Analytical Models," AIAA Paper 85-0779, 26th Structures, Structural Dynamics and Materials Conference, Orlando, Florida, April 1985.

5. Crawley, E.F., and Aubert, A.C., "Identification of Nonlinear Structural Elements by Force-State Mapping," AIAA Paper 84-0992-CP, presented at the 25th Structures, Structural Dynamics and Materials Conference, Palm Springs, CA, May 1984, AIAA Journal, vol. 24, Jan. 1986, pp. 155-162.

6. Felippa, C.A., and Park, K.C., "Direct Time Integration Methods in Nonlinear Structural Dynamics," Computer Methods in Applied Mechanics and Engineering 17/18 (1979) pp 277-313, North-Holland Publishing Company.

7. Kitagawa, M., and Kubomura, K., "Transient Load Analysis Method for Large Linear Structures with Local Nonlinearities and its Application to Space Structure Payload Load Analysis," AIAA Paper 85-0734, 26th Structures, Structural Dynamics and Materials Conference, Orlando, Florida, April 1985.

Friday, June 21 - MULTI-BODY DYNAMICS

Questions and Answers Following: "Applications of Multi-body Dynamics to Space Structures" by Keto Soosar

Gerald Goudreau, Lawrence Livermore National Laboratory: Before we settle on one code, at least a year is needed to give all parties a chance to solve some benchmark problems. I would encourage the task of defining some benchmark problems which results in middle-to-large size structural models and which are portable fashion so that all interested parties can use them. Let me add that I don't think of the modes coming from a structural program as fulfilling the portability criterion. Rather, the basic finite element model on a magnetic tape with full formating definition fulfills the portability criterion.

Soosaar: I agree that it's difficult to get useful benchmark problems into the research community. You can begin with extremely simple problems but they won't address system level complexities. And when you get the system level complexity that tends to overwhelm the researcher. We have to pursue large problems in the research community because this will force others to think about them. The question of where a model is physically valid is appropriate. There are many situations where it won't be.

Ed Haug, University of Iowa: Keto, I'd like to agree with you and make two points. First, we need to do a lot of work, but it's not clear to me that a single all purpose code is the answer. I think a multidisciplinary activity which provides cross-talk between codes is required. The second point that I would like to raise, (and this is self criticism), is that the effects of geometric nonlinearity of these multi-body systems is child's play compared to the nonlinear problems of the hydro-code discipline where nonlinear material behavior is treated. I don't think the multi-body area is technically difficult, but we need to move out and get some things done.

Soosaar: Allow me to address your two points. First, I agree that there should not be just one code. The situation is similar to that of elastic finite element codes which have tended to evolve according to industry needs. There are certain finite element codes that are very appropiate for either civil engineers, nuclear engineers, mechanical engineers or aircraft and spacecraft

PRECEDING PAGE BLANK NOT FILMED

engineers. I'd like to see a trend towards a more generic encompassing formulation, but, at the user level branch off into different types of application; one for robotics, one for the auto industry, another for aerospace. Unless of course efficiency is served by having them together. The formulations need to be in the same class. The second point is on structural nonlinearity in the multibody problem not being as severe as in other fields. I generally agree, but these effects do play a roll in structural control. There the controllers tend to be extremely sensitive to anything that isn't linear or simple. This has to be included in a number of cases and, unfortunately, those are also the ones where you have to deal with large multi-degree-of-freedom systems, so you get bitten there. But it's a point well taken.

J. M. Housner: We're going to have to cut short the questions at this time and, hopefully, during the panel discussion we can get back to some of these.

Questions and Answers Following: "Applications of Multi-Body Dynamics to Mechanisms and Robotics" by Prof. Steve Dubowsky.

Chung: How do you apply a boundary condition in the finite element analysis for the hydraulically driven actuator?

Dubowsky: The hydraulic actuators provide loads on the system and they're not boundary conditions. We model in the dynamics of the hydraulic actuators as we do with other dynamic controls. In the control systems, you typically have dynamics, you have transfer functions in there which are, in fact, dynamics and they yield additional dynamic equations which need to be assembled with the dynamic equations of the structure and the manipulator. What happens is the actuator forces appear in the generalized force terms. Their values are a function of, in control system jargon, the states of the system. So there are additional dynamic equations involving the state variables and forces which interact with the structure. Much like the structure states appear in the dynamic equations of the actuator and the control system.

Ramen Singh, DYNACS Engineering Co., Inc.: You reference modal synthesis for coordinate reduction. If you have motion at the joint, how do you separate the vibrational coordinate from the joint coordinate?

Dubowsky: How do you separate the vibrational coordinate?

Singh: Yes, you said that you do modal synthesis to reduce the coordinates.

Dubowsky: That's right, of the individual links.

Singh: Of the individual links?

Dubowsky: That's right, of the individual links, not of the entire system. That way we can describe the links in great detail and yet not deal with so many degrees of freedom when we come up to the system level.

Joe Padavon, University of Akron: How would you handle the friction in a structural joint in an adverse environment, say like in space?

Dubowsky: Right, we do put friction in; we model the bearings; we model the compliance of the bearings. Essentially, what we do is have degrees-of-freedom at each of the joints. The relative rotation at the joints are degrees-of-freedom in the system. So the friction, much like the control forces, become torques on those joints. We retain joint forces and moments in directions of degrees-of-freedom of the joints. We have to. That's where the actuators are and that's exactly the way the friction interacts with the system. Those terms appear in the generalized force terms. We do not eliminate those.

Padovan: I understand that, but what happens if you lock-up and you go through rather large deformations in the rather flexible arms. Then you're going to have changing stiffnesses for the joints. How would you handle that situation?

Dubowsky: I'm sorry. Please repeat the question.

Padovan: If the manipulator arm were to experience an adverse lock-up, it would then experience some kind of an adverse bend, with resulting stiffness changes on your links. How is this handled?

Dubowsky: I think your question is, if you had sufficiently large structural deformations within the individual links, such that the mass or stiffness matrices change, how would that be handled? We don't consider that case because we assume that the links will not bend much. Maybe somebody wants to look into that. When you are dealing with very flexible manipulators that may be something worth looking into. I should point out, though, that we don't have the watershed solution in modeling of flexible manipulators. You'll hear from other speakers who are doing relevant work with different approaches.

John Hedgepeth: One of the things that some individuals are considering for robots, or manipulator arms, is to feed back the joint angle with very tight control. In that way, persumably, you would take out the effect of the compressibility of the hydraulic fluid, for instance, in your example. And the interesting thing, to me, would be, (a) what your results would look like and (b) once having done so, and being confident at being able to do so, shape your acceleration and deceleration transients in some way that produces less jerk.

Dubowsky: The vast majority of manipulators feed back their joint angles and joint velocities. Some even feed back end point information. And one of the principal researchers in controlling flexible manipulators thinks that's one of the keys to this problem. That's Bob Cannon at Stanford. So all manipulators feed back joint position. If you tried to raise the bandwidth of those joint control systems so that they could, in fact, control the flexibility.

John Hedgepeth: I didn't say that.

Dubowsky: If you try to raise the bandwidth of that control system, raise the gains, to make it a very tight control system, you may go unstable because you'd have put the structural resonances within the bandwidth.

John Hedgepeth: I'm talking about tightening the loop within the joint itself.

Dobowsky: That's right. If given a flexible manipulator with high performance, high bandwidth, high gain control systems on the joints to control those angles very precisely, one that doesn't get rid of the arm bending, does it get rid of the hydraulic compliance? It does, but you can't do that. At least most people don't know how to do it. There may be a way to do it. I don't think I know how to do it without getting very clever. The reason is because you're trying to close the control around a compliance, around a resonance, and if you try to put that resonance within the bandwidth of the control system in order to control it, you wind up with real stability problems.

John Hedgepeth: I could be wrong because I'm certainly not an expert in this area, but my knowledge is that if you control the degree-of-freedom that you're measuring and you're doing it within a tight loop, what is external to that does not infringe on the stability of that loop.

Dubowsky: Well, that's in part true, but the basic problem is that the hydraulic compliance is within the loop of what you're trying to control. You're trying to control itself through its compliance so the compliance is within that loop and the phase shift from that compliance comes in. Classical conventional control theory will not enable you to do that.

Questions and Answers Following: "Modeling and Formulation" by Jim Turner

Gerald Goudreau, Lawrence Livermore National Laboratory: You mentioned the Lagrange multiplier method led to sparser system equations. Do any of these multi-body codes try to exploit that sparsity in solving the system equations?

Jim Turner: Yes. DISCOS does exploit that. There's no reason to compute a lot of things if you know they're going to come out to be zero and so you can incorporate these things into the code. In our own deployment code, it's absolutely crucial. There are huge blocks of equations that are zero. So, obviously, you're not going to store all those zeros, you want to take advantage of some of the ideas that have been used in structural dynamics for minimizing your storage and minimizing your mathematical operations that lead to null results.

Goudreau: In terms of this dual formulation that you were describing at the end of your talk--between Lagrange methods and the reduced order methods--DISCOS handles both formulations? Is that what you are saying?

Turner: DISCOS handles the Lagrange multiplier. It is not set up to reduce out the degrees-of-freedom. It deals with the maximum dimension of the problem at the acceleration level. That's because it treats each individual body as being separate. At the kinematics level, it deals with the constraints across the hinges. It then is able to provide correct differential equations at the kinematics level so that you're not carrying the unnecessary degrees of freedom at that level.

Goudreau: So it reflects the righthand methodology that you are comparing there as opposed to the reduced relative degrees-of-freedom?

Turner: When you get the relative coordinates you have the minimum degrees of freedom. That is not what DISCOS is doing.

Goudreau: OK, thank you. That's what I really wanted to know.

Harold P. Frisch,NASA Goddard Space Flight Center: I think you brought up a real good point in showing the formulation between the Euler formulation versus Lagrange formulation. I think it brings up the point that you really need a multitude of multi-body formulations geared toward specific problems. The DISCOS code was motivated by spacecraft problems where there are very few constraints between joints. Frequently, bodies are tied together with 6 by 6 stiffness matrices. So we have basically a zero constraint, zero rigid constraints between the bodies. On the bottom line of the DISCOS formulation, you're inverting a matrix which is the size of the number of constraints. In the Euler approach, there the matrix that you are inverting is the mass matrix. Whereas if you have a multi-body problem with lots of constraints between bodies, then you're inverting a matrix which is the size of the number of degrees-of-freedom. Could you comment on whether you feel that we should go to one large code that solves all problems or very specialized codes that solve their specific class of problems very efficiently?

Turner: The latter.

Frisch: OK, good.

Ramen Singh, DYNACS Engineering Co., Inc.: I have one comment to make on what Harry just said, I believe he is alluding to inverting the sparse mass matrix which is of large dimension but contains very few non-zero entries. This sparse character is lost when constraints are maintained for any large relative motion across the hinge. If you have an n by n holonomic constraint matrix, which is differentiated twice and substituted into the equation of motion or applied through Lagrange multipliers then, in maintaining the constraints, you have to use some iterative method to achieve solution accuracy. The trade-off is between inverting a small sparse matrix versus stabilizing the constraints.

Turner: Well, there is an ad hoc technique....

Singh: All kinds of ad hoc techniques.

Turner: I won't elevate it any higher than that. Carl Bodley resolved this difficulty by treating errors at the acceleration level of the Lagrange multiplier calculations as though they were noise in the momentum. This noise is corrected at the kinematics level of calculations by computing an impulse momentum correction that makes the velocity state consistent with the constraints. This doesn't resolve the difficulty for holonomic constraints, but it does enhance the confidence level in the calculated results.

Martin Tong, The Aerospace Corporation: When you apply that procedure are you not introducing external disturbances to your system, thereby changing the system momentum? Whereas when you use the Lagrangian multipliers to couple the holonomic constraints into the problem, you haven't. When you start introducing impulse type of corrections are you not changing the angular or the linear momentum of the system?

Turner: I'm sure that's true, but I think you're making the problem much less severe. The large dimension of the Lagrange multiplier matrix itself, I think, is what is beginning to degrade the accuracy of the solution. You'll see this in dual spin spacecraft. If you don't have that impulse momentum correction operating at each integration call of the derivative routine, constraint forces or torques which should be zero start to grow. After a time, the numerical simulation has numbers in the 10^{-4} range that should be near zero, 10^{-16}.

Tong: Isn't it true then that when that occurs, your simulation is already deviating from what is the truth?

Turner: What you're doing is correcting in the hypertangent plane to the constraint surface. You're still going to have some numerical errors. The hope is to delay the time at which it becomes meaningless.

Tong: Thank you.

Questions and Answers Following: Dynamics of Articulated Structures by Edward J. Haug

John M. Hedgepeth: I want to compliment you on your talk. I'm in total agreement with you that one should look at the problem of the individual elements statically since that may be enough, as your door example showed. Also, if you first take care of the static degrees-of-freedom and satisfy static equilibrium and compatibility, a very solid basis is provided on which dynamic effects can be added. As to your code which will be commercially available, I'd like to know from whom, and when?

Haug: The code will be available from a small company called Computer Aided Design Software, Inc. (CADSI) in Coleville, Iowa. If you write to me, I can give you the information, or get the information to them, after which they'll contact you. It will also be available from Computer Vision Corp. Presently only the rigid body version is available. The flexible body version should be ready by October 1985. We have to close the financing arrangement at the end of this month to be able to add additional staff to complete the commercialization. It takes a lot of money and time. The code is operational now and we have examples manuals; I can guarantee you we can have it in your hands by October. I'll guarantee that even if I have to do the work myself.

Jim Robinson, NASA Langley: On your door example, how much worse off would you have been if you had simply applied a static deformation of opening the door and used one load case with the kinematic constraints of simply supported boundaries. What displacement pattern would you get when using one static shape instead of five?

Haug: Jim, I didn't try anything like that, but I think that roughly what happens is that as the door swings, the boundary conditions change, and it isn't a one-parameter change in the boundary condition. It's a five-parameter set of boundary conditions. So I think you have to get all five in there.

Robinson: There's another method where you use deformed shapes with inertias from the first load. Next you orthogonalize, (as in the Gram-Shmidt method), and essentially create another shape function. The procedure is similar to what Joe Kinney used to call the general method of structures where an applied unit load is used to remove redundancy. This requires that you identify redundancies before initating the solution.

Haug: Right. By the way, we didn't invent this technique. We borrowed it from the substructuring area.

Ted Belytschko, Northwestern University: Just a simple question. Could you explain to people in the structural dynamics community what you mean by a static mode?

Haug: That term was coined, I presume, by Craig and his student Chang. They referred to shapes due to static unit loads, calling such shapes, static modes. They also have shapes due to selected unit deformations. I forget the name they gave to those modes. It's a fairly descriptive term which appears in Craig's book, and Chang's thesis.

Belytschko: There's a counterpart method that was published about 5 years ago by Wilson. Just this morning one of the attendees pointed out to me that it's actually a Lanczos type vector. The large number of different nomenclatures can get confusing.

Haug: I claim nothing new. We're just trying to put it all together.

Housner: Right, that's why we have this mixed community here to straighten that all out. Bob Melosh, please.

Robert Melosh: Ed, would you give us some details on the variable step size algorithm you're using? In particular, what orders you're dealing with and how much drift there is between one order and another during the integration process.

Haug: Sure, Bob. Actually, we use a standard code, that Champine and Gordon wrote. It's called DE. It's the most commonly used ordinary differential equation solver, I think, in the world. The code, I think, goes from first to seventh order. I don't believe it goes higher than seventh and it adjusts the order of the algorithm and the step size in a predicter-corrector operation to satisfy error constraints. It's standard Adams-Moulton initial value problems. Basically, the theory is in Champine and Gordon's book that's 10 years old. You can find the FORTRAN listing of the code in the book. It's a super code though. I say it's super, but I tell you one of the problems with this kind algorithm. Yesterday the structure experts talked about implicit/explicit, I have trouble because I don't think that terminology is completely consistent with the initial value problems literature. I think that they are doing some things to filter out some high frequency content whereas the algorithm we use doesn't. If there's a high frequency mode in there, it's going to work hard trying to find it. And the computer bill can just be terribly high. I have a feeling we need to do dual rate integration methods or, at least, some frequency discrimination and get some of this high frequency stuff out of there because it's very expensive. So, Bob, I hope that answered your question.

Ramen Singh, DYNACS Engineering Co., Inc.: In one of your charts, you had nine computations which were like preprocess and the weighting of simulation code.

Haug: Right.

Singh: Would you comment on some of those computations involving quadratic terms, second order terms, in modal coordinates?

Haug: I like to think about these deformation modes as defining the kinematics of the deformation field. Now once I've defined the kinematically admissable deformation fields, then I go through the formulation of my equations of motion and, indeed, coupling occurs. Now I suspect that those quadratic terms represent the effect of the coupling between the gross motion and the deformation. I simply view them as a mathematical fact of life.

Singh: No. I meant the quadratic terms or second order terms in modal coordinates themselves?

Haug: We did have a student who dropped them out in his thesis and he did some sensitivity studies and found that they did have a moderate influence. I suspect you could drop them most of the time and not lose anything. I'm not sure. We basically just kept everything because I don't think it cost us that much to do so.

Harry Frisch: You really can't drop those terms out. The best example to illustrate how you would get yourself in trouble by dropping that quadratic term out is, if you have a spinning spacecraft with a flexible boom along the spin axis, that quadratic term tells you how the moment of inertia of the system changes.

Haug: But there is a danger in including some of those and not all of them. That's what we've found.

Frisch: You're in deep trouble if you drop any of them out.

Questions and Answers Following: Model Reduction Strategies for Interconnected Flexible Body Simulation by G. Mann

Martin Tong, The Aerospace Corporation: I think the idea of reducing the model is very good, but in this case I think it is successful because of at least one condition. It hinges on the fact that the system mass matrix is time invariant. So for time varying systems, you might need other approaches.

Man: Agreed. Actually for this system, the system mass matrix is time varying because we are dealing with a dual spinner.

Tong: I was just commenting. Two things can happen. Your spinner is an axisymmetric body.

Mann: It is not that axisymmetric.

Tong: Oh it is not?

Man: If you noticed in an earlier slide, the rotor is not shaped like a nicely shaped cylinder, it has many appendages sticking out. It is not that symmetric. Very complicated, very complex.

Tong: So does this method apply to a time varying mass matrix?

Mann: Well, what we do is that we perform the modal influence analysis by freezing the rotor with respect to the stater. And we have to do it for many positions. After we have gone through that process, then we know exactly what modes we should retain. But, by no means is that final because we still do not have the spinning effect in. That's why we need a multibody simulation tool, to put the spinning dynamics back into the picture for design verification.

Jimmy Ho, Lockheed: I'd like to make the following comment, especially when you talk about flexibility, i.e., the modal truncation. In a lot of structures like that you don't really talk about modal truncation, you really should concentrate on modal selection. In a configuration like that we had associated with JPL on the flight experiment for a big antenna flown out of the Shuttle cargo bay. We received a lot of modes from JPL. Many of these have very low frequency but yet never get excited. In this specific configuration, which is nearly round, there exist modes involving normal deflection and those involving inplane deflection. They are functions of the radial parameters and also the angle. The angle could be sine or cosine of m theta, the m can be 1, 2, 3, 4; that is, cyclic harmonics. But this cyclic harmonic, if you integrate, just like you integrate modal phi into capital phi then you will find out the higher order cyclic harmonic like two thetas or three thetas, have a zero resultant. Now those modes never get excited and yet they are there. Boundary conditions play a strong role in selecting modes which are excited. In other words, modal characteristics will determine your selection. That is really a very important practice.

Man: I agree with you. Simply truncating frequency does not work well. Especially when you deal with one component at a time. Consider, for example, Galileo. If you just look at the free-free mode of the stater structure, that is the connecting part between the rotor and the platform, you might truncate the higher frequencies above 50 hertz. When you hook the various components back together within DISCOS, you can compute the combined system mode and compare that with the NASTRAN combined system mode. They turn out to be extremely different. It is not easy to know which component modes are important till you get to the system level.

Questions and Answers Following: Computational Aspects of Multi-body Dynamics by K. C. Park

Housner: Any questions or comments for Dr. Park? Looks like everybody is in 100% agreement with you.

Questions and Answers Following: Constraint Elimination in Dynamical Systems by R. P. Singh

Ed Haug, University of Iowa: Raman, I wanted to comment that we have used this method substantially over the past couple of years and I think it is optimally stable. I think you esentially cannot define a better set of generalized coordinates, but just two quick comments. One, if your "A" matrix is time dependent then technically you should redo the singular value decomposition. There is always the uncertainty as to how much error you get if you do not do that. Second, the place where this thing is just absolutely beautiful is when you get into near singular or terribly ill-conditioned configurations of mechanisms. I suspect on some of the deployable structures, we get into very ill-conditioned situations and I would strongly recommend the use of a technique like this.

Bahram Nour-Omid, Lockheed Palo Alto: The way I understand singular value decomposition especially for finding ranks of matrices, it's like breaking a hazel nut with a sledge hammer. I think there are better ways of solving this problem, namely methods based on QR factorization. The computations are a lot more efficient than singular value decomposition. Have you made any comparisons with any QR factorization or stable forms of QR factorization?

Singh: Yes, and I will recommend to you to a paper by Dr. Jung and Pinson. The comparisons have been made there. There is no computational efficiency compared to SVD method.

Nour-Omid: If you are having a sparse matrix, you can take advantage of the sparsity of the matrix in the QR factorization a lot better than in singular value decomposition.

Housner: Can we get a clarification on your response to that last question?

Singh: I have not tried to find the QR decomposition of "A" transpose "A" or "A" itself.

Nour-Omid: All you do is the QR decomposition of "A," you never go to the normalized equation "A" transpose "A." You can do the QR decomposition of "A" itself in a very sparse fashion. I think Michael Heath at Oak Ridge National Laboratory as well as Alan George have shown that the QR decomposition is very stable. If you read their work in the recent literature, you will find that there are very nice and efficient ways of achieving that without getting any form of instability in the problem.

Singh: That will be something to look at.

Jer-Nan Juang, NASA Langley: I can comment on this. If you are very interested you should read the paper by Klema and Laub in the IEEE Automatic Control Journal about 1982 or 83. They have done substantial studies on SVD including those kinds of factorizations, etc. Their conclusion is the SVD is better than anything else. That is their conclusion, not my conclusion. I just suggest you read that.

Editors Note: Klema, V. C. and Laub, A. J.: The Singular Value Decomposition: Its Computation and Some Applications. IEEE Transactions on Automatic Controls, Vol. AC 25, No. 2, April 1980, pp. 164-176.

Housner: By better, do you mean stable but not necessarily a minimum number of computations? I believe that in Gaussian elimination there is a minimum number of computations.

Singh: SVD method does give you a very good determination of rank, but the gentleman is pointing out that the new work, which I am not familiar with, provides a very acurate measure of the eigenvalues and, if that is the case, and the computations are less, then definitely it may be a fine thing, but we have looked at the methods using "A" transpose "A" and working on a transpose "A". OK, so that's something new which I would like to learn more about. But it is usually the practice that you do not want to find the eigenvalues because the accuracy of finding the singular values is much better than in the accuracy of the eigenvalues for rank determination. The QR algorithm is something of a new algorithm which is very accurate for Eigenvalue determination then maybe that is something which should be looked into.

K. C. Park, Lockheed Palo Alto: What size of a problem are you advocating this for? Are you implying that this could be used for equation systems exceeding say several hundred degrees-of-freedom with varying mass matrix?

Singh: I have not seen cases being run with any code with thousands of variables. We have been planning, and have implemented some of it successfully, using more than one CPU on a super mini 32-bit machine with two CPU's. There you can have larger dimensions by partioning the tasks with the system matrix and the constraint being worked on by one CPU and assembling the equations on another CPU. We have found that we have employed something like 120 or 220 variables. We have gone nowhere near a thousand variables. Maybe that will be the trend of the future where you do work with thousands of degrees-of-freedom. But I have no experience with that size.

Park: From what you have experienced so far, do you advocate this type of approach for large system equation sets also?

Singh: I would recommend its use on large systems equations only when the decomposition subroutines are worked to suit the particular problem you are solving or the class of problem you are solving where there are many zeros and you do multiply the algorithm when it does not take n^2 times n number of computations. Our example had a pointing requirement which was half an arc second, we have tried Newton-Raphson stabilization on the constraint equation and we had too much computational effort. As a matter of fact, we paid a penalty, and maybe we did not know how to use the method correctly. Nevertheless, we achieved better computational speed with singular value decomposition.

Questions and Answers Following: Nonlinear Characteristics of Joints as Elements of Multi-body Dynamic Systems by E. F. Crawley

Gerald Goudreau: Two questions about your examples. In the truss where you show the insertion of your joint, was that an axial spring and dash pot?

Crawley: No, I'm sorry that was sort of a model of a truss in flexure and then the pieces of the truss ended up being represented by a beam in flexure and the joints were rotational. They were pinned with rotational freedom.

Goudreau: You are calling the whole structure a truss and, therefore, the major points of the truss are pin connected so there are no moments being transmitted beyond the two adjacent beams. It was a frame? Each member is a bending beam?

Crawley: I wish I could get that picture, that truss back. Actually what I was doing was modeling, if you can look at my fingers here, I was modeling two frames of a truss which come together and are connected by a connector on the top and the bottom which actually have axial play.

Goudreau: So each of them are axial members.

Crawley: Correct.

Goudreau: To give the overall bending.

Crawley: The net is to influence the overall bending.

Goudreau: OK, that clarifies it.

Crawley: That was just a "back-of-the-envelope" calculation.

Goudreau: Now, you showed another slide with three hysteresis loops measured, called axial tension, transverse bending, and inplane bending. In those bending ones, are those moments?

Crawley: That's right. That is someone else's result. In that case it was a sleeve connected joint, I believe, and they did a test this way, and then they did a test that way.

Goudreau: Oh I see, so each one of those is the response in one direction to the an excitation in the same direction.

Crawley: That is correct.

Goudreau: I guess what I am leading up to asking is whether experimentally you are to the point of characterizing the multiple variable state.

Crawley: There are obviously all those cross coupling effects too.

Goudreau: Is anybody trying to do that experimentally, sort of characterize the whole response surface?

Crawley: We have an analytic extention of our method that allows those possibilities. But I do not know anyone yet who has experimentally tried it.

Goudreau: I'm wondering whether it is worth putting both these complex formulas together for a one-dimensional model.

Crawley: Well these are the elements of the joint. You have to do this before you can do the cross-coupled one.

Goudreau: Thank you.

John Hedgpeth: Ed, I'd like to compliment you on your talk. I'm certainly in agreement that there is so much unknown about what we are talking about here that embarking on expensive and detailed analyses of particular joints particularly to try to estimate their nonlinearity and damping is probably not fruitful at this time. However I would say that...nor is it particularly worrisome at this time about making an exact identification of the dynamic characteristics of the joint. I'd like to report on the fact that the state of the art on actual design of joints is such that we designers--no matter whether we are dynamists or not, and incidentally I have designed and flown joints in space--we designers think that when we design the joint for the purposes for which it is intended, that joint is stiff enough. The definition of stiff enough depends on what the particular application is. I guess that my concern here is that I think that we are worried here about something which is a couple of steps beyond what our real problem is today. Our real problem today is the paucity of any experimental data whatsoever on joints which are the kind of joints which would be considered to be candidates for flight, and I am talking about projects like Space Station and large deployable reflector and others for which every time one looks at these structures, one's first question is well, can you build a joint so that we can make the whole structure work? We need data and we need the kind of data that you would turn the joint over to the laboratory technician and say give me a force-deflection curve, even without giving us cycling on it, just give me some data to begin with because there is a great deal of over optimism on the part of people who construct joints and actually use them in structures as to what their actual structural characteristics are. And in some cases, the data that I have seen are just appalling in how bad that joint is as a piece of structure.

Crawley: Let me reflect on those comments. I agree with you that there is a marked paucity on any joints that are proposed to go into the pieces of Space Station. This is part of an effort to generate that data and understanding. Furthermore, I would point out, John, that additional instrumentation that is necessary to get the more complete set of data other than doing just the traditional four-stroke measurement is very minor. What I'm saying is that if you send down a joint to XYZ test lab with the addition of an accelerometer to the test setup, they can in fact produce a much more useful body of data than if they just do a four-stroke measurement. So, I think we are in essential agreement there. You actually said two different things, one was that as a designer you think the joints are stiff enough, but the other was that there is over optimism about how stiff joints are. These are almost contradictory remarks. I would say that the role of the analyst in this situation is to try and produce the analytical tools to assess how much difference is enough so that the designers can use those tools. How much difference is enough in the design of the joints for the power tower. That is a question that at least two or three groups of people around the country would like to know the answer to.

MULTI-BODY DYNAMICS PANEL DISCUSSION

COMMENT, J. M. HOUSNER, NASA LANGLEY: I've placed a list of questions on the table before you, and I have also put them on the screen. Some of the questions deal with items that arose during the entire 3 days of the workshop. Whom are we trying to please? Who are the customers of the CSM activity? Is CSM meant for software development or for researchers in the development of methods and algorithms? Questions were also raised concerning whether methods that will be developed by CSM would be "ivory tower" methods and if they would be good only for producing one or two technical papers but would be of no real practical use. If that is true, how can it be avoided? And, of course, the issue of concurrent processing looms in the background. In this workshop, there has been some discussion about benchmark problems; I'd like to consider that idea again. In some respects, in order to achieve some of the goals that the CSM activity has set before itself, modularity seems to come into play in addition to the ability to check out, confirm, or investigate research given particular analytical methodology. The last item to consider is whether there should be a follow-up workshop to this one.

COMMENT, BARNA A. SZABO, WASHINGTON UNIVERSITY: I benefited a great deal from participating in this conference, and I would, in particular, like to refer to the focus problem of the first day because it provided a basis to talk about something specific. I was impressed by the efforts of the Lockheed representatives in making these analyses. They assigned a high-quality analyst to focus on the problem, and I found that they spent a large amount of time, something on the order of 6 months, to do that particular problem. Dr. Beckman commented that if complex analyses are to be performed in the design industry, they need to consume 6 hours rather than 6 months of an engineer's time. So basically we're speaking about a very large gap between what is possible to do today and what the real world expects us to do. Consider the problem this way. On the horizontal axis you would plot the time, and on the vertical axis you would plot the cost. Every project given a fixed problem, such as the focus panel, would have a time/cost trade-off curve. That curve would be shaped perhaps like the letter J. How much it would cost to do this particular problem really depends on the technology. What we are facing here is a technology gap between what is being delivered today and what the aircraft industry would like to do. It seems to me that in the first two items in the CSM focus, the first thing should be to define the goals. What is it that we would like to achieve? Methods and algorithms and/or software development should be subordinate to those goals. If our goal is to be able to do the focus panel reliably (in an afternoon) by a trained analyst, I think we will clearly identify the problem difficulties that we have to face. I believe that kind of performance increase is a possibility today even though we would have to explore and put forth a little more effort to accomplish it.

QUESTION, J. M. HOUSNER: Do we have any further comments?

COMMENT, K. C. PARK, LOCKHEED PALO ALTO RESEARCH LABORATORY: I'd like to clarify the behind-the-scenes story associated with the focus problem. If we had to redo that analysis again, it would certainly not take 6 months. It took 6 months because we wanted to cover as much ground as possible. We were searching everything: correct software, element formulation, shell theory, solution procedures, and information pattern. In normal day-to-day analysis situations, you would not need that kind of thoroughness; therefore, we would not need that much time.

QUESTION FROM THE AUDIENCE: How long would you need?

COMMENT, K. C. PARK: I would say we would need about a week for preparing the initial data, another week for checking out the initial data and doing some linear and buckling analysis, and another week, or at most 2 weeks, to do linear analysis. It's a matter of a month rather than 6 months.

COMMENT, J. M. HOUSNER: Thank you for that clarification. Would anyone like to discuss the issue of how to support the development of methods to insure that those methods could be of practical nature rather than just of academic interest?

COMMENT, HAROLD P. FRISCH, NASA GODDARD: We talked about benchmark problems, problems of academic interest and methodologies given by academia. Most benchmark problems, at least at Goddard, are oriented toward project support. We're developing both NBOD, that I wrote, and DISCOS, which we put together. These developments were driven by needs. We got tired of deriving equations over and over again, and we needed to get the job done efficiently and reliably. Working in a project support environment is invaluable in getting these programs checked out and developed and getting the capability that you need. It's almost a natural filter to take out what is really not that important. You're going to develop the capability that you need over and over again. In the academia problems, you say, "I'll do those next year," and next year never comes.

QUESTION, J. M. HOUSNER: What you're saying is to let the identified applications drive the focus or benchmark problems?

ANSWER, HAROLD P. FRISCH: Yes.

COMMENT, ROBERT MELOSH: As an academician, I'd like to comment on that idea. I think the role of the academician, the university, or the research-oriented person is quite different from the role of the production analyst. Our role, as I see it, is one of creating and trying ideas, not necessarily proving them. There is a development phase that has to be gone through. Ideas have to be rejected as well as tried, but I don't think its necessarily the responsibility of the researcher to develop the idea. He may not be the right person to do that kind of work. He may not know computer programming very well. In fact, most codes developed at universities are inefficient codes. That's natural because the researchers are not concerned with efficiency as much as dealing with the ideas.

COMMENT, JIMMY HO, LOCKHEED: It's really nice to get everybody to become more interested in flexible multi-body dynamics. I have personally experienced in the past 10 or 15 years of activity that traditionally we have a lot of structures people, and at the same time we don't have many controls people here. Controls people are working on these problems too, but somehow there's a gap in the middle, and this gap is flexible multi-body dynamics. At Lockheed, I was in charge of the program to bring these two disciplines together. I am using the flexible multi-body dynamics to bridge the gap between the structures and controls disciplines. I think this is the proper approach. I've talked to a lot of my colleagues at different companies, and we all have the same problem. I hope this meeting becomes a trend for the future.

Another comment I would like to make is that previously when we did a simulation, we were caught at the end of the hardware development program. Management said, "I want a multi-body spacecraft with this kind of configuration." Then they have a cartoon drawn and get some analyses performed. At the end of this process we are called in. The simulation is only used for performance analysis. Actually it has another more important function. If used at the early stage of the iterative design process, it can be used as a design tool to influence the design. That is really important because it can be used to perform trade-off studies. There's an overall design strategy, and I think this is the basic function of the multi-body dynamics discipline.

COMMENT, J. M. HOUSNER: Does the panel have any response to the comment concerning the use of multi-body dynamics? I gather what Jimmy Ho is saying is that multi-body dynamics can be a bridge between controls and structural dynamics used early in the design.

COMMENT, KETO SOOSAAR: I think we have a problem of accessibility to the multi-body codes which is generally quite severe. Most people who would like to use phase codes are designing a system at the front end and would be completely intimidated by their unfamiliarity with the archaic terminology and methodology within the multi-body community. I think there is probably the need for some simplified, first-order tools for helping build the bridge. Also, when you're dealing with systems that have tremendous amounts of control needed (some very severe large space structures have that) the phenomenology of multi-body is a relatively small contributor to the pure structural response. When you're getting close to the actual performance, and you're really trying to squeeze down three or four orders of magnitude of response reduction, if you haven't taken the multi-body effects into account, your control system will be unstable. The point is that we need to educate more people to be sensitive to multi-body dynamics. We need to have tools that are more than a simple lumped mass with a single, one-mode flexible appendage on it. We need some analytical tools that can help people understand them, the issues, and the interfaces.

COMMENT, HAROLD FRISCH: One of the research tools that we need more than anything else is the ability to get flexible body data from the structures program into the multi-body program. Currently, we have the Space Station coming up, and it's inconceivable that we'll have one flexible body model for the entire Space Station. Structures people will use substructure analysis, and we'll have substructure models of the Space Station. We do not have the ability to get even the rigid-body mass properties from a substructure analysis out of a NASTRAN code. We need to spend some effort in educating the structures people to provide the ability to pass information easily out of their code into the system so that the multi-body people can transform that data into the data they need. In particular, you need a low mass distribution and a grid point location. You can get lump masses easily, but it's almost impossible to get grid point locations out of something like NASTRAN. We have to spend some effort in developing computational linkages out of your structural dynamics into a format so that the multi-body people can pick the data up and transform it into the various coupling coefficients they need to do their work.

COMMENT, KETO SOOSAAR: I think one of the problems we have to face in the development of these tools is that most such tools require sufficiently large capitalization, yet they tend to be used by a large distribution of organizations, none of which could capitalize such a tool themselves. As a result, if you have a lot of small tools being developed, you really don't have over a long period of time any feeling for their validity. One of the points I keep trying to make is that there should be a strong government institution, nonprofit institution, or governmental lab, like NASA, that continues the development, not just of multi-body tools but also of the sort of things that you just referred to that the Space Station needs. Otherwise we're going to wind up with a whole bunch of half-done jobs, the validity of which will be in question. There will be a larger and larger gulf between the one or two large companies that can afford to build them and academia, which is trying to solve those problems. We need to get the DISCOS', if you will, to academia occasionally. We need to get them interested in what NASTRAN improvements need to be made.

COMMENT, JOE PADOVAN: I'd like to make two remarks. First of all, I think a lot of information on stiffness and mass matrices is probably already available on NASTRAN, just through a straightforward DMAP process in which you could just down load most of that information to a file. The problem is making it compatible with your own data structure. Let me comment on what was just stated a moment ago. It's literally impossible for the standard academic to get into a real code. MARC is proprietary to any research application. The only thing we can really do in academic settings is to use MARC on an educational basis or NASTRAN. I venture to say that's true of all the major codes today. We have no way of getting to source codes. If we have to respond to real world code development in academia, it means we have to hawk our own little code and to reinvent the wheel every other month or every other year. We need access to the mainframe code, a national code (which I don't necessarily like). What we do develop goes directly out into the market and can be benchmarked. It's very hard for us to sell a little home-brewed code to get wide attention.

COMMENT, K. C. PARK: Let me clarify something concerning academia versus industry or the ideal research versus the day-to-day problem solutions. Although I am in industry, I'm in a research area. I think the middle ground has paid off, that is, the ideal situation is to let researchers develop and explore their ideas and to construct a software environment that would accelerate the translation of their ideas into the production software environment. This concept has been proved quite productive, at least in our laboratory where we have been working on a system called NICE, for which Carlos Fellipa has been the prime architect. We are happy to be sponsored by NASA under the CSM project and hope by next April to have the NICE system available to the entire community. This would provide a data management system to which modules could be attached so that academia can access the code and contribute to their new ideas in the form of a software module. Thus, other people could share in the multi-body dynamics initiative of Lockheed under the joint sponsorship of NASA. Whatever we do under that project would also be available under this CSM/NICE effort. Be just a little bit patient; in about a year we hope to distribute that system.

COMMENT, GUY MAN, JPL: I have two comments about universities versus industry. I think there is a place for both parties in the multi-body simulation world. What we need to define now are the goals. What are we trying to shoot for in the development of any software system? Are we trying to gain speed? Are we trying to make the multi-body simulation program modular so that you can pluck something out and then replace it with hardware for testing? How complicated do you want to make the system? It's usually dictated by technology development needs. We have to understand what kind of projects are on the horizon that need multi-body simulation program support. Once we understand that, let's sit down and define what is lacking before we initiate the multi-body program and research. The university environment is an ideal environment for coming up with ideas once we have identified the problem areas to focus on. For example, how do you introduce damping into the system? How do you validate that? When you develop a very complicated code, do you have an absolute yardstick to check it? Validation is a big problem. What kind of test should you design to verify your software? I'm not in favor of checking code against code. We have to check code against tests. Have we defined those? Some people have, but they are not known to the general public. So let's focus on goals and requirements. What are the problems we face? Is it computing speed that we try to crank up? What do we eventually use these multi-body programs for? Is it for just the design of the control system or for the testing and validation of a very complex system such as a space platform where we have complicated onboard software and the whole structure vibrating, moving all over the place, with sensors and actuators mounted everywhere.

How do you keep track of where they are? How do you validate your control design? Are the current programs fast enough for us to do that?

COMMENT, J. M. HOUSNER: I tend to agree with you that there's certainly a lack of tests in most disciplines and especially in the multi-body discipline. We've seen that need in the general areas of space application, mechanisms, and robotics. Also, one of the large roles played by the multi-body dynamics area, besides some of the things that you mentioned such as controls and stability, is the production of loads that are due to dynamics. Those loads as output from multi-body analysis become the input for detailed finite-element structural analysis. That was the case when I worked on the Space Station. I was asked questions about the loads coming off the remote manipulation system. Without a multi-body analysis, the best you can do is take a good guess based on back-of-the-envelope calculations. So, the loads question is one which multi-body dynamics analysis can answer.

QUESTION, R. J. HAYDUK, NASA LANGLEY: Yesterday, Bob Melosh identified two groups that are present here and that we could consider customers of the technology that we are attempting to develop. One group was industry application specialists; the other was the computer code developers who are marketing their codes and their technology. Linking the university group, university researchers, and the CSM group together as a team, which of the two customers do you think we should be trying to serve, the industry application specialist or the computer code developer?

COMMENT, K. C. PARK: I would hope we would be able to serve both. For those of you who are interested in software engineering and who have had a chance to read one of Carlos Felippa's papers, we made that point very clear. The software environment that we are striving to develop is to serve both ends--the application end and the research environment. That is a very very difficult objective. We have reason to believe we have succeeded partially in that we have input from people who do nothing but day-to-day analysis. We have algorithm developers (I should partially qualify myself for that since I have succeeded in developing algorithms for our test bed), and we have Gary Stanley and Carlos Felippa who are excellent software experts. They also manage to use and test their software to improve and to extend the software environment itself, so we have broad, although admittedly limited experience. It's going to take about 3 to 5 years to evolve as a mature software system.

QUESTION, R. J. HAYDUK: How do you expect the other computer software entrepreneurs to take advantage of this system that you're talking about? Do you expect someone from MARC Analysis Research to come in and try your system here at Langley?

ANSWER, K. C. PARK: We are certainly not restricting any potential users because it will be an open software system a year from now. As long as the U.S. government does not object, others can get it. In other words, if users do not intend to export that system to another country that is our adversary, I have no reason to believe they would not get permission. We cannot give permission. We will deliver it to NASA, and it's up to NASA to decide whether a party will have access to it. But, as far as we are concerned, it is a completely open system.

COMMENT, JIM TURNER, CAMBRIDGE RESEARCH: The thing that's going to catch people's attention is your ability to attack, perhaps, a special class of test problems for which you could demonstrate that you have something that will have substantial improvements over what's available in industry. On the industrial side, the people doing research and code development are usually one and the same. If you're going to impress people, I think you've got to demonstrate that you can get either efficiency or reliability or something else over the presently available tools, or people are just not going to notice.

QUESTION, MARTIN TONG, THE AEROSPACE CORP.: Dr. Park, what is this system which Lockheed is trying to make open to industry or the public?

ANSWER, K. C. PARK: Jeff Stroud probably can clarify contractual aspects better than I can, but as far as we are concerned, we deliver source codes, not just the absolute, to NASA.

QUESTION, MARTIN TONG: Do you mean the source code of the multi-body simulation?

ANSWER, K. C. PARK: No, I mean the NICE system. The NICE system is a software environment associated with a data base management system. We did Gary Stanley's shell analysis results using a module that we have developed for NICE. We have developed a large space structure transient dynamic program module and several other modules for internal in-house use, but multi-body dynamics is a module that we just started. We hope to have that module available to the community in about 2 1/2 to 3 years.

COMMENT, W. J. STROUD, NASA LANGLEY: I'd better add a few comments about the test bed. The idea of this test bed system originated out of a frustration at having to do a substantial amount of coding, often developing a new code when a researcher wanted to do research on structural analysis methods. This was particularly true in university research. The resulting methods were usually code-dependent. When a code dies (a student graduates) those methods are, for the most part, lost. To address both those problems -- unnecessary software development and ineffective technology transfer -- we wanted to develop what we call a test bed system. We're working on that, with Lockheed Palo Alto, right now. The system will have hooks, if you wish, so that applications-specific modules can be plugged into this test bed. If a university person wanted to develop some analysis module, he wouldn't have to develop everything; there would be a system available for him to work with. The disadvantage is that a university person would be constrained to have his module work on that system. The advantage is that he would not have to develop his own software system. It's a two-edged sword.

We would also be using this test bed system to find out the ingredients of a modern structural analysis software system and how those ingredients should fit together to determine how the data should be passed around, and what might be a higher-order language that could be used with it. Certainly it would be used to test out applications modules. We're doing the work right now on a VAX 11/780. In a few months it will be a VAX 11/785. The VAX system was chosen because Lockheed Palo Alto and almost everybody has a VAX or access to one, whereas if we used our Control Data system at Langley, it might not be so easy for people to use. We want to move the test bed toward multiple processor computers.

We have a contract with Lockheed Palo Alto, where NICE was developed by Carlos Felippa. K. C. Park is part of the Lockheed team that is working with us. Among other things, that contract puts NICE in the public domain. Right now I guess we're the only ones, other than Lockheed, that have the system. Our original intent, and we hope we can carry through with this, is to have even another test bed system in addition to the NICE-based system and to transfer both to industry and universities to evaluate, to comment on, and to give us some feedback. Then we would make some changes. The test bed might be a combination of the two test beds before it's over.

Right now we only have the NICE test bed and that might be what we end up with. We would hope that we will all be able to make use of the NICE test bed system.

QUESTION, GERALD GOUDREAU, LAWRENCE LIVERMORE NATIONAL LABORATORY: Is the NICE specifications document available now so that we could get an early look at what it will be like? Are you still trying to define the requirements or what you would expect them to be?

COMMENT, W. J. STROUD: We're working on the documentation, aren't we K. C.?

COMMENT, K. C. PARK: Carlos Felippa or Gary Stanley should have been here to answer these questions because I'm not really a software specialist. As far as I know Lockheed made an agreement with NASA to make the NICE system public. Now whether that implies that it is immediately available to the community as a whole or we wait until we deliver the test bed system to NASA, I don't know. I would have to ask my boss. My impression is that it is up to NASA to distribute it because Lockheed would not be responsible for distributing the software.

COMMENT, W. J. STROUD: That's about right. One never really has software that's fully documented and that's a problem. It's also a little bit premature right now to be sending out documentation. What we're doing is coupling the NICE system with the structural analysis analyzer, SPAR, which is also in the public domain. We're studying how to do that right now.

COMMENT, GERALD GOUDREAU: My question was really what does NICE contain, what is the software environment, and what are the analysis tools? You've got an intermediate grey area of utilities and things like that so I'm not talking about complete documentation but what you expect the software system to contain besides the data management system.

COMMENT, W. J. STROUD: NICE contains no analysis capability.

QUESTION, GERALD GOUDREAU: And no analytical utilities?

ANSWER, W. J. STROUD: We're coupling SPAR with NICE to give a linear structural analysis capability for now. We're taking it in steps.

COMMENT, GERALD GOUDREAU: I think of grey areas to be things like equation solvers, eigen packages, etc. You call that structural analysis modules rather than environment utilities.

COMMENT, W. J. STROUD: SPAR contains equation solvers and eigensolvers. NICE does not.

QUESTION, JOE PADOVAN, UNIVERSITY OF AKRON: I don't want to knock a VAX 11/785 but it is a rather slow machine, and many of the nonlinear long-run problems that run on a VAX 11/785 may be very cumbersome. Is there any anticipation of moving that software system up to a higher level machine?

ANSWER, W. J. STROUD: Certainly. We have a problem at Langley and perhaps other places that are using computers that are beginning to have a similar problem. For the first time, our computer center is saturated. We recently bought a CYBER 205 supercomputer from Control Data, and we thought it was going to be many months, maybe even a year, before we were saturated again. However, we were saturated in three to four months. For that reason and portability we went to VAX. Certainly, the test bed will move to other computers, because we know that computationally intensive problems will have to be on computers larger than a VAX 11/785.

QUESTION, JOE PADOVAN: Has it been arranged friendly enough so that the process of falling off the VAX to, say, an IBM or a higher machine would not be a major problem?

ANSWER, W. J. STROUD: We're going to go to UNIX. Right now we're using a VMS VAX, and we will have to go to some type of a UNIX-based operating system for portability. That's a trade-off right there. We'll probably be investigating what we should be doing to make it transportable. The flex computer that we will be buying is a Unix machine.

QUESTION, HARRY FRISCH: Could I ask a multi-body question? K. C. and Jim, you've both brought out the multi-body problem associated with deployment dynamics. In deployment dynamics we have many many bodies tied together all doing a specific thing. The question I'm asking, after hearing Dave Benson's presentation, is if you think deployment dynamics should be carried out in a multi-body environment or are you better off to use Dave Benson's work? Where do you think there's a trade-off?

QUESTION, JIM TURNER: I didn't hear Dave's presentation. Dave, was your research based on PDE (partial differential equation) descriptions or nonlinear constituent laws?

ANSWER, DAVE BENSON, LAWRENCE LIVERMORE NATIONAL LABORATORY: The presentation I made yesterday was about putting rigid-body dynamics (among other things) into DYNA 3-D, which is an explicit finite-element code. We have plans to do essentially the same thing to NICE. One of the things that Harry was interested in is the possibility of taking nonlinear finite-element codes, using them to define these large antennas that are going to be deployed, and relaxing the degrees of freedom between beam elements so that they're not actually a structure anymore (that is, tying only the translational degrees of freedom and relaxing the rotational degrees of freedom). Then we could potentially do these deployment problems with a mixture of that kind of technique combined with rigid bodies.

COMMENT, JIM TURNER: My comment would be, not having seen everything that Dave's done, that it sounds as if you can accomplish the same thing through two different approaches. I'd have to see the formulations to really speak with any confidence. I don't see why the multi-body approach would be terribly penalized. He would be modeling with one element or several elements per rod where we would just have a rigid link with an inertia, a mass, and some geometry associated with it. I don't see why necessarily one would have a distinct advantage over the other.

COMMENT, HARRY FRISCH: It's a good area to put some money into. I think one of the things we do need is an efficient method of studying deployment dynamics. I don't think we have it today. I see Dave's code as a potential for a quick solution. I'm just wondering whether the multi-body dynamics approach is going to carry along too many terms and be just computationally inefficient.

COMMENT, JIM TURNER: One of the things that we would advocate, and I think it addresses one of the questions that Jerry brought up in the beginning, is to have truth models no matter whether it's deployment or just simply a multi-body code without deployment as a part of the process. I think that in reality for practical engineering you need to be able to have the ability to go in and dynamically linearize or alter the structure of the equations so that you can bring down the levels of complexity. But you still have to retain a truth model. The idea of embracing a rigid-body model as being the truth is wrong. You still have to be worried that you may have a structure control interaction that drives you unstable, and if you have a rigid-body model you're not likely to predict that. I think you've got to be able to simplify the models, and that should be part of the computer code package. You still have to retain the truth.

COMMENT, K. C. PARK: Let me confess one thing before I comment on that aspect. This is the second time I've had a really tough time giving my presentation, this is mainly because I did not have any results, and those of you who have had any experience presenting a talk when you do not have any results, know it's terrible. Now getting on to the issue of deployment dynamic problems, I believe that the successful software would embody both capabilities within the same topology; that is a finite element. You can do finite-element analysis, structural dynamics analysis and stability analysis. If you want you should be able to do nothing more than rigid-body dynamics simulation. If you want to mix it, you should be able to mix it. That is a real challenge. If you take brute force finite-element formulations, you can solve those problems, and I and my colleague estimated how much computer resources it would take if we went all the way with brute force finite-element formulations. For that 100 meter reflector antenna it would take a Cray XMP 2 1/2 weeks to do a decent deployment simulation. If you use DISCOS right now, I believe it's beyond the DISCOS capability.

COMMENT, HARRY FRISCH: I totally agree with that statement. DISCOS was never intended for that type of problem. DISCOS was intended for a few coupled flexible bodies. Once you get much beyond 10 or 15 bodies you're just getting out of the realm of practicability.

COMMENT, J. M. HOUSNER: We've had some good discussion on these critical subject areas; I know we could probably discuss more. I want to thank the speakers, panelists, and the audience that has been so attentive, enthusiastic, and participated so well. I'd also like to thank the CSM group who worked so hard to put this meeting together, especially Jeff Stroud.

CLOSING COMMENTS, W. J. STROUD: I'll repeat what Jerry Housner said; thank you very much. If there are some questions, issues, or suggestions that you feel need to be brought out in the next couple of weeks, please bring them to our attention. We are going to be putting together the final proceedings. You already have the preliminary proceedings. Thank you very very much for coming.

National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA CP-3034, Part 2

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Computational Methods for Structural Mechanics and Dynamics

5. Report Date
May 1989

6. Performing Organization Code

7. Author(s)
W. Jefferson Stroud, Jerrold M. Housner, John A. Tanner, and Robert J. Hayduk, Editors

8. Performing Organization Report No.
L-16560

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, VA 23665-5225

10. Work Unit No.
505-63-01-10

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

13. Type of Report and Period Covered
Conference Publication

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The conference publication contains the proceedings of the Workshop on Computational Methods for Structural Mechanics and Dynamics held at NASA Langley Research Center, June 19-21, 1985. The Workshop was organized into the following four sessions:

(1) Local/Global Nonlinear Stress Analysis
(2) Tire Modeling
(3) Transient Dynamics
(4) Multibody Dynamics

Transcriptions of discussions are also included.

17. Key Words (Suggested by Author(s))
Local/Global
Parallel Processing
Global/Local
Tire Modeling
Transient Dynamics
Multibody Dynamics

18. Distribution Statement
Unclassified - Unlimited

Subject Category - 39

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	252	A12

NASA-Langley, 1989

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171